Keele
UNIVERSITY

Study on automatic citation screening in systematic reviews: reporting, reproducibility and complexity

by
Babatunde Kazeem Olorisade

A thesis submitted in partial fulfilment of the requirements for the award of the degree of
DOCTOR OF PHILOSOPHY

KEELE UNIVERSITY
MARCH 2019

Abstract

Background: Challenges in the conduct of systematic reviews have led to research into and development of support tools targeting the process or specific stages. There is a growing body of research into the use of text mining methods for citation screening support. However, these studies are reported with insufficient details to support reproducibility and technical comprehensibility of the models.

Aim: To investigate transparency in the reporting of citation screening in systematic reviews particularly as it relates to reproducibility and technical comprehensibility of the models.

Method: A literature review was conducted to investigate the methods being used for citation screening support and the type of information reported about them. Consequently, a reproducibility assessment of studies was undertaken to systematically assess the level of reproducibility of the studies and the factors responsible. This was followed by two studies to investigate the structural complexity of the models being used. A text mining based tool was developed to support citation screening and tool support research.

Results: The review showed a growing body of research but a lack of technical information about models and reproducibility enabling information. The reproducibility assessment identified information essential to study reproduction and suggested a checklist. The complexity assessment and feature enrichment studies reinforced the need for complexity related information in study reports. The citation screening tool demonstrated how a tool can be useful for both practice and research.

Conclusions: Research into text mining based tool support for citation screening in systematic reviews is growing. The field has not experienced much independent validation. It is anticipated that more transparency in studies will increase reproducibility and in-depth understanding leading to the maturation of the field. The citation screen tool presented aims to support research transparency, reproducibility and timely evolution of sustainable tools.

Acknowledgements

I would particularly like to thank the duo of Prof. Peter Andras and Prof. Pearl Brereton for their supervision of this work. Your constant motivation, encouragement and advice are invaluable. I would also like to thank Prof. Barbara Kitchenham for her help whenever her professional advice is sought on the project and for providing one of the datasets used in this project. I would also like to thank Dr Ed. de Quincey for his support during the conduct of the literature review in this work.

I am particularly grateful to the National Information Technology Development Agency (NITDA) for the award of the PhD scholarship through its NITDEF scholarship scheme, to conduct the research activities that lead to the results presented in this thesis.

Deserving appreciation, are all the staff of the Computing and Mathematics department at Keele University. Thanks to all the teaching, technical and administrative staff who have helped even in the smallest way. I would like to thank all the past and present PhD students of the department with whom I have crossed paths. Thank you all for your friendship and contributions.

Special thanks must go to my loving wife - Rasheedat Yahaya and adorable daughters - Kareemah and Aneesah for your support, motivation and endurance during the course of the study; One thousand and one words are not enough to express my love for you. I would like to thank my friends who have been of immense support during the project. I would like to thank my brothers and sisters for their support and prayer. On a final note, I will like to thank my late parents - Osuolale Olorisade and Idayat Olorisade (nee Raimi), who both toiled day and night to set me on the right path, may the almighty 'Allaah' forgive your sins (Ameen).

Declaration of authorship

Whilst registered as a candidate for the above degree, I have not been registered for any other research award. The results and conclusions embodied in this thesis are the work of the named candidate and have not been submitted for any other academic award.

Disseminations

In the course of the PhD, work reported in this thesis was published and presented at a number of conferences. Details of the papers that were prepared for publication, including the conference and seminar activity, are presented in this section.

Journal articles

Olorisade, B. K., Brereton, P. and Andras, P. (2017). 'Reproducibility of studies on text mining for citation screening in systematic reviews: Evaluation and checklist'. In: *Journal of biomedical informatics* 73, pp. 113.

Conference papers

Olorisade, B. K., Brereton, P. and Andras, P. (2017). 'Reproducibility in Machine Learning Based Studies: An Example of Text Mining'. In: *Reproducibility in Machine Learning workshop at the 34th International Conference on Machine Learning*. Sydney, Australia.

Olorisade, B. K., Brereton, P. and Andras, P. (2017). 'Reporting Statistical Validity and Model Complexity in Machine Learning based Computational Studies'. In: *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*. ACM, pp. 128133.

Olorisade, B. K., Quincey, E. de et al. (2016). 'A critical analysis of studies that address the use of text mining for citation screening in systematic reviews'. In: *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*. ACM, p. 14.

Presentations

External talks

$34th$ International Conference on Machine Learning (ICML 2017), *Reproducibility in Machine Learning workshop*, Sydney, Australia, July 2017.

$21st$ International Conference on Evaluation and Assessment in Software Engineering (EASE 2017), Karlskrona, Sweden, June 2017.

$20th$ International Conference on Evaluation and Assessment in Software Engineering (EASE 2016), Limerick, Ireland, June 2016.

Internal talks

TeMACS: A transparent tool for automatic citation screening in systematic reviews, $8th$ *Computing Postgraduate Research Day*, Keele University, April 2018.

Reporting statistical validity and model complexity in computational studies, $2nd$ *Faculty of Natural Science Postgraduate Symposium*, Keele University, May 2017.

Reporting statistical validity and model complexity in computational studies, $7th$ *Computing Postgraduate Research Day*, Keele University, April 2017.

Text Mining Based Support System for Citation Screening in Systematic Review, $1st$ *Faculty of Natural Science Postgraduate Symposium*, Keele University, June 2016.

Three-minutes thesis (3MT), *Institute of Liberal Arts and Sciences (ILAS) Postgraduate Conference*, Keele University, April 2016.

A critical analysis of studies that address the use of text mining for citation screening in systematic reviews, $6th$ *Computing Postgraduate Research Day*, Keele University, April 2016.

Automatic citation screening in systematic reviews using text mining based techniques, $5th$ *Computing Postgraduate Research Day*, Keele University, April 2015.

Posters

Reproducibility in Machine Learning Based Studies: An Example of Text Mining, $34th$ *International Conference on Machine Learning (ICML 2017), Reproducibility in*

Machine Learning workshop, Sydney, Australia, July 2017.

Automation of the citation screening stage in systematic review, *Institute of Liberal Arts and Sciences (ILAS) Postgraduate Conference*, Keele University, April 2016.

Contents

List of Figures

List of Tables

List of Abbreviations

AHRQ Agency for Healthcare Research and Quality.

BD2K Big Data to Knowledge.

BOW Bag-of-Words.

CLEF Conference and Labs of the Evaluation Forum.

CS Citation Screening.

CV Cross Validation.

DERP Drug Evaluation Review Program.

DM Data Mining.

EASE Evaluation and Assessment in Software Engineering.

EBSE Evidence-Based Software Engineering.

EPC Evidence-based Practice Center.

ESEM Empirical Software Engineering and Measurement.

FAIR Findable, Accessible, Interoperable and Reusable.

HTML Hypertext Markup Language.

http Hypertext Transfer Protocol.

IE Information Extraction.

IR Information Retrieval.

KDD Knowledge Discovery in Databases.

LDA Latent Dirichlet Allocation.

LSI Latent Semantic Indexing.

MCC Matthews Correlation Coefficient.

MDL Minimum Description Length.

MeSH Medical Subject Heading.

ML Machine Learning.

NIH trans-National Institute of Health.

NLP Natural Language Processing.

NTCIR NII Testbeds and Community for Information access Research.

PCA Principal Component Analysis.

PMID Pubmed Identification.

SE Software Engineering.

SGML Standard Generalized Markup Language.

SQL Structured Query Language.

SR Systematic Review.

SV Support Vector.

SVD Singular Value Decomposition.

SVM Support Vector Machine.

tf term frequency.

tf-idf term frequency-inverse document frequency.

TM Text Mining.

TREC Text REtrieval Conference.

URL Universal Resource Locator.

WSS Work Saved over Sampling.

XML eXtended Markup Language.

Glossary

Some of the terminologies used in this work has been derived from multiple disciplines. The glossary does not represent formal definition but an attempt to quickly familiarise the reader to the concepts.

class — A set of specific target output attributes/groups/categories of the data.

classifier — Refers to a model created through the implementation of a classification algorithm, that maps input data to a set of output categories.

conclusion validity — This type of limitation to the validity of a study concerns the relaibility of the conclusions of the study.

construct validity — This type of validity limitation raises questions on how well the design choices for the study is able to address the research questions.

external validity — This concerns limitations regarding whether the outcome of a study can be genaralised to other situations.

feature — In text mining, this is a feature (or term) refers to an individual distinct word/text unit that collectively compose the body of text.

fit — The process of creating a simplified representation (model) of a dataset in a way that it can be generally used successfully given new data. For example, to identify data similar to the one it was trained on.

internal validity — This is a type of validity limitation introduced by other bias factors or conduct of the study.

learning — The mastering of the underlying distribution/pattern of a dataset and the mapping of each data input to some target attribute.

model — An abstract representation of the real distribution of a dataset. It is the artefact produced as a result of the training process.

negative class — Name used to represent the output category of no interest in a dataset during classification.

over-fitting — Describes a situation when the model too close to the reality. That is, it ended up learning both the detail and the noise in its training data to the detriment of its performance on new data.

pickle — A module in Python that implements binary protocols for serializing and de-serializing a Python object structure. Pickling in Python refers to the process of converting an object to byte stream for storage purposes and un-pickling is the opposite..

positive class — Name used to represent the output category of interest in a dataset during classification..

predict — Describes the process where a trained machine learning model attempt to suggest the class of a new data given its underlying dataset's classes.

scrape (scraping/web scraping) — The process of using automatic software to gather specific (textual) information from websites.

stemming — The process of removing inflection (suffixes, prefixes and affixes) from words to return them to their original root (stem) forms.

training — The process of providing a machine learning model with a set of data to learn from.

weight — The value assigned to data points (or class of data) during weighting.

weighting — The process of assigning a multiplication 'cost' to data points in machine learning to empasize the importance or otherwise of the data (sometimes used as class, term, feature or sample weighting).

CHAPTER 1

Introduction

This thesis presents a set of studies undertaken to investigate experimental transparency in studies reporting the use of text mining techniques for automatic citation screening in systematic reviews in relation to reproducibility and understanding of complexity of the text mining models. The work leads to the development of a 'transparent' text mining based tool to support citation screening in systematic reviews and set the stage for cross-team research collaboration.

This chapter describes the focus of the thesis, which concerns experimental transparency, reproducibility and structural complexity of the models used in text mining studies relating to automatic citation screening in systematic reviews. Reproducibility in this work implies the ability to reproduce study results through the replication of their processes while complexity refers to the hypothesis the techniques utilize in making the classification decisions. A brief introduction to evidence-based software engineering, the systematic reviews process with an emphasis on citation screening and the challenges of the method are provided. The research questions of the work are outlined and the work's motivation and objectives are explained. The novelty of the thesis and its contribution to knowledge are pointed out. The chapter ends with an outline of the structure of the thesis.

1.1 Background

A brief introduction to Evidence-Based Software Engineering (EBSE) and the Systematic Review (SR) process is presented in this section. The section also contains a highlight of the SR process.

1.1.1 Introduction to EBSE

Evidence-based research and practice was initially adopted and has been successfully practised in medicine (Sackett, Rosenberg, Gray, Haynes, & Richardson, 1996; McKibbon, 1998; Reynolds, 2008). The success has lead to its adoption in other research

areas like nursing, sociology, education, psychology etc. (Kitchenham et al., 2009; on Evidence-Based Practice, 2006) and eventually Software Engineering (SE) (Kitchenham, Dyba, & Jorgensen, 2004).

In 2004, Kitchenham et al. (2004), suggested an evidence-based research approach, EBSE, to the SE community in order to bridge the gap between research and practice. The objective of EBSE is to make available, empirical evidence to assist practitioners make informed decisions when adopting SE techniques and practices (Dyba, Kitchenham, & Jorgensen, 2005). In consistency with evidence-based medicine, EBSE requires the following five steps to execute (Dyba et al., 2005):

i) Convert a relevant problem or information need into an answerable question.

ii) Search the literature for the best available evidence to answer the question.

iii) Critically appraise the evidence for its validity, impact, and applicability.

iv) Integrate the appraised evidence with practical experience and the customer's values and circumstances to make decisions about practice.

v) Evaluate performance and seek ways to improve it.

Steps 'ii' and 'iii' are achievable through a methodical review of the literature - SR. EBSE is therefore anchored on SR for the gathering of the right evidence. A detailed report on how to conduct SRs in the context of SE is published in Kitchenham et al. (2004) and updated in Kitchenham and Charters (2007), Kitchenham, Budgen, and Brereton (2015).

Since its introduction and adoption, EBSE has continue to grow considerably in different topics of SE. In a tertiary study covering the use of SR in SE between 2004 and 2008, 20 unique studies were found by (Kitchenham et al., 2009) with additional 33 in (Kitchenham et al., 2010) over the same period. By extending the tertiary study search date to 2009, Da Silva et al. (2011) found an additional 67 studies. This, in addition to at least two annual conferences with emphasis on empirical research and preference for SRs - the international conference on Evaluation and Assessment in Software Engineering (EASE) and the international conference on Empirical Software Engineering and Measurement (ESEM), are indicative that EBSE (and invariably SR) has become an intrinsic part of SE research.

1.1.2 Introduction to systematic reviews

Literature review is an integral component of every research activity. Traditionally, the task of searching, gathering and reviewing of literature are conducted in an ad-hoc way. This approach exposed the practice to at least two identifiable flaws:

i) exhaustive coverage of existing literature is not guaranteed.

ii) the process is usually not repeatable.

The poor quality of narrative reviews increased the quest for more formal methods for producing a systematic and explicit way to provide up-to-date evidence on a subject of interest. SR (also sometimes referred to as systematic literature review (SLR)), is a literature review approach that provides a rigorous, dependable and 'auditable' review methodology with the main goal of building an impartial and complete synthesis of available empirical research evidence on a specific topic; thus, creating a focused platform on which practically useful decisions and conclusions can be made (Kitchenham et al., 2004; Kitchenham et al., 2015; Higgins & Green, 2011).

SR process consists of three major phases: planning, execution and documentation (Kitchenham et al., 2004; Kitchenham & Charters, 2007; Kitchenham et al., 2015). These phases are further divided into stages. The phases, constituent stages and the interaction between each of the stages is shown in Figure 1.1.

1.1.3 Systematic review process

The SR is conducted following a laid out approach decided before the process is commenced. Figure 1.1 shows the three phases of the SR and 10 stages of activities in the phase (Kitchenham & Charters, 2007). The stages of each phase are briefly discussed in this section.

1.1.3.1 Planning phase

The goal of the planning phase is the production of a 'protocol' - a priori laid down plan on how the review process will be conducted, candidate studies judged and research questions to be answered by the review outcome. The planning involves three stages:

i) research question: The first stage is the definition of questions that will provide a direction to the need of the SR, the construction of the string for document search and the types of data required to satisfy the inquiry (Kitchenham et al., 2015; Kitchenham & Charters, 2007).

ii) protocol development: the second stage is the development of a review protocol. The protocol contains detailed definition of the process to be adhered to during the review. This includes outlining the approach to undertake while searching and selecting the candidate studies, conditions to be met by each study for consideration in the review, the data to be extracted from each study, assessment criteria, study allocation to reviewers etc.

iii) protocol validation: This includes running a pilot review to test the understanding and relevance of the protocol prior to review scale application. This may lead to the revision of the protocol. The protocol can be revised at any stage when any inadequacy is identified.

1.1.3.2 Execution phase

The guidelines specified in the planning phase are applied to the five tasks identified for the execution phase. So, once a version agreed to by all members of the review team is available the execution phase can commence. The five stages involved are:

i study identification: The first stage of the execution phase is to identify candidate studies using the search strategy defined in the protocol. The coverage of every possible research likely to be relevant to the review is key to the success of this stage (Kitchenham et al., 2015; Kitchenham & Charters, 2007).

ii citation screening/study selection: The second stage of the execution phase is the filtering of relevant studies from the outcome of the identification stage applying the criteria for inclusion and exclusion predefined in the 'protocol'. This is normally conducted in two steps: first, by removing the clearly irrelevant studies based on the content of their titles and abstracts while the second one is by reading the full content of the remaining studies to determine if they are actually relevant based on the conditions set out in the inclusion/exclusion criteria. For the sake of reliability, it is recommended that each article be screened by at least two reviewers with a chance for resolution over any disagreement.

iii study evaluation: Following an agreement by the review team on a set of qualified studies, their quality are assessed based on predefined criteria.

iv data extraction: Information in each study that qualifies for each data item defined in the extraction form are extracted.

v data synthesis: The concluding stage of the execution phase is the collation and aggregation of the extracted data with the intent of answering the research questions.

1.1.3.3 Reporting phase

Once the review process is concluded, then it is time to formally document and report all the processes and outcomes.

i) define strategy: At this stage the reviewers may define how the report is to be written and organised in consonance with the protocol and the research question. Further suggestions on the possible structure and contents of the report can be found in (Kitchenham et al., 2015; Kitchenham & Charters, 2007).

ii) reporting and validation: The final stage is to formally write the review report and possibly have it validated by an independent researcher.

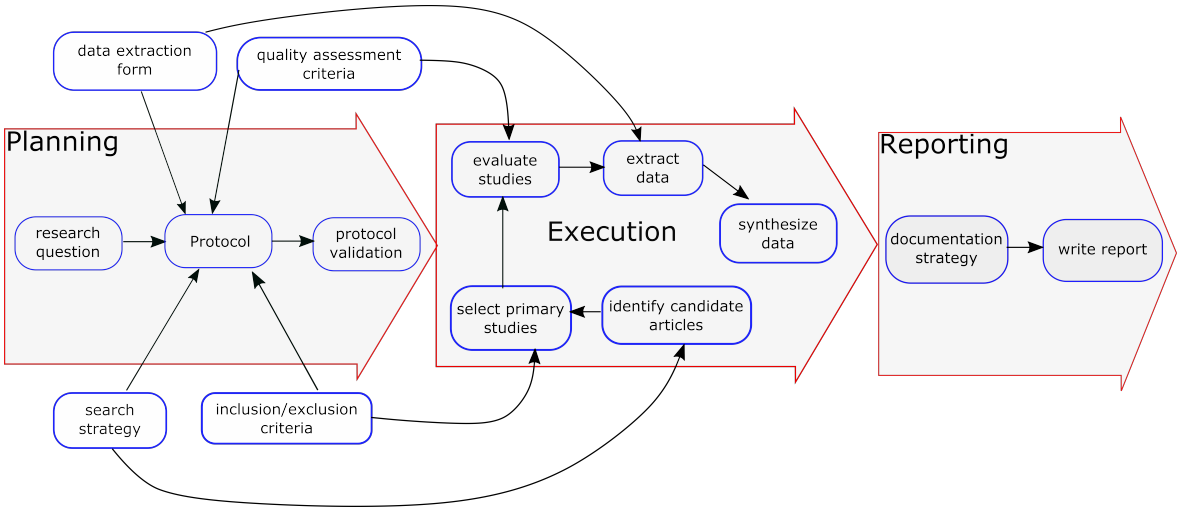

Figure 1.1: Systematic review process with task interactions

1.1.4 Systematic review experience in SE

The growing interest in empirical research and SR in particular has lead to useful feedback on the use of the SR guidelines proposed by Kitchenham (2004). Several researches have reported the guidelines as fit for purpose and have suggested areas in need of improvement based on their experiences, expertise and needs (Staples & Niazi, 2007; Brereton, Kitchenham, Budgen, Turner, & Khalil, 2007; Dyba, Dingsoyr, & Hanssen, 2007; Riaz, Sulayman, Salleh, & Mendes, 2010; Turner, Kitchenham, Budgen, & Brereton, 2008). An alternative guideline for SR in SE was proposed by Biolchini, Mian, Natali, and Travassos (2005).

A tertiary study by Kitchenham and Brereton (2013) identified about 18 areas in need of improvement suggested across experience studies. Some of these improvement suggestions include procedural improvements while some advocate for tool or other forms of external support. The challenge posed by the amount of time and effort consumed by conducting a SR is common among the experiences (Riaz et al., 2010; Babar & Zhang, 2009; Zhang & Babar, 2013; Petersen, Feldt, Mujtaba, & Mattsson, 2008; Brereton et al., 2007; Carver, Hassler, Hernandes, & Kraft, 2013). Some studies have favoured the need for a tool to support the whole (design, conduct and reporting) SR process (Zhang & Babar, 2013).

These drawbacks have positioned the processes involved in the conduct of the SR as prime candidates for automated support tools. Staples and Niazi (2007) believe success in the automation may be achieved faster by targeting individual stages rather than the whole process. Data extraction, study selection and data synthesis have featured as the top areas in need of automated support (Staples & Niazi, 2007; Hassler, Carver, Kraft, & Hale, 2014; Z. Yu, Kraft, & Menzies, 2016). Study identification is another area that has been identified to be manual and labour intensive that could benefit from automated tools (Carver et al., 2013).

SRs are conducted to gather evidence to improve the body of knowledge of any particular subject, therefore, they are often repeatedly conducted at intervals to update existing knowledge. The current rate of publications and the time it takes to complete a review may make the findings of a review become out-dated quickly. On average, a medical review takes one year from protocol development to publication (Borah, Brown, Capers, & Kaiser, 2017). So, a tool that preserves the states (of each stage) of the previous review and update the states with new review exercise is potentially useful at reducing repeated activities and updating findings.

A number of studies have been published on the subject of automated tools to support SR resulting in a range of tools. One of the prevailing approaches in the automation is the use of Machine Learning (ML) techniques, a computational method, through Text Mining (TM). The Citation Screening (CS) stage is one of those that had attracted the greatest interest in terms of applying TM techniques for SR support. More details on the current efforts at providing automated support for the whole process and individual stage of the SR is discussed in Section 3.2. O'Mara-Eves, Thomas, McNaught, Miwa, and Ananiadou (2015) reviewed 44 articles on the application of TM to support the CS stage of the SR published between 2006 - 2014. An additional 12 articles were found covering 2015 - 2018 (see Section 3.5). Only five tangible tools have so far evolved from the studies reported in the articles - ABSTRACKR (Wallace, Small, Brodley, Lau, & Trikalinos, 2012), Gapscreener (W. Yu et al., 2008), SWIFT-Review (Howard et al., 2016), Rayyan (Khabsa, Elmagarmid, Ilyas, Hammady, & Ouzzani, 2016; Ouzzani, Hammady, Fedorowicz, & Elmagarmid, 2016) and Fastread (Z. Yu et al., 2016).

Further issues associated with the use of computational methods, TM in this case is to provide automated support tool for SR processes particularly the CS stage will be further discussed in the next section.

1.1.5 Research motivation

Reporting scientific experiments in a way that the results can be understood and independently reproduced is a standard requirement of scientific reporting. It is however difficult to ensure computational experiments are well communicated and reproducible (Goecks, Nekrutenko, & Taylor, 2010). The application of computational methods for building support tools for the conduct of the CS has its own drawbacks particularly in effectively reporting and communicating the research process to others. In an attempt to provide solution to some of the identified SR drawbacks through automation with TM models, the computational methods employed lead to other issues like research reproducibility and transparency, understanding and dealing with the models' complexity among others. Complexity in the ML context is considered in terms of the comparative relation between the data size, the feature

vector size (see Section 2.2.2) and the size of the classification algorithm's hypothesis space (Joachims, 1998). The lower the complexity of a model the better it has learnt to generalize over the dataset. The larger the size of a dataset the better the learning (chances) of a ML model and (possibly) the lower the model's complexity.

The tools mentioned in Section 1.1.4 and studies reviewed in Section 3.3 have shown the potential of TM techniques to improve reviewers' experience in the CS stage of the SR and possibly the quality of the review outcome. Specifically by reducing the amount of reviewer time and effort spent selecting relevant studies and reducing human bias applicable to this stage. Despite these advantages, the tools and the 56 studies reviewed in Chapter 3 on the use of TM techniques to provide automated support for the CS stage of SR have shown a lack of experimental transparency. Independent researchers due to the multiple complex procedures of the tools and methods are unable to independently reproduce the results or replicate the processes. Consequently, the ability of independent researchers to understand the complexity of the models which may be used to interpret their performance and propose changes or improvements to these existing tools and methods is limited.

Reproducibility of experimental outcomes is a key phase of scientific enquiry, which provides the foundation for understanding, integrating and extending existing results towards new discovery (Goecks et al., 2010). In a similar way, 'good reporting' has been reported to be an essential component to future research development (Miguel et al., 2014). Thus, good reporting and reproducibility are critical to knowledge advancement and a shorter new discovery turnaround in an evolving field. Only two of the tools have been reported independently evaluated, ABSTRACKR (Rathbone, Hoffmann, & Glasziou, 2015; Gates, Johnson, & Hartling, 2018) and Rayyan (Olofsson et al., 2017; Couban, 2016); none of the studies have their results independently reproduced.

There have been many studies published on the use of TM techniques for automatic CS. The field and awareness of the potential of this method is growing, experimental datasets are becoming more accessible thus the number and potentials of studies are rapidly increasing. But all these are still with low levels of collaborative research and independently reproduced results. There is an absence of study replication and insufficient technical details in reports. Therefore an in-depth investigation into the usefulness of these discovery evolution and enabling issues would be beneficial to the research community.

1.2 Research Objectives

The overarching goal of this thesis is to investigate aspects of the quality of reports in studies on automatic tools for CS in SR using TM techniques. Particularly, how the information provided in the reports support reproducibility and understanding

of the quality of the models being reported vis-á-vis model complexity. The specific objectives of the thesis are to investigate:

i) transparency in TM based CS studies based on the level of information provided on the experimental procedures and the resulting models.

ii) the conditions for the reproducibility of the study results and how this is satisfied by the studies.

iii) the need for reporting the complexity details of the TM models.

Three research questions were developed to guide the focus of this project:

RQ1: What information is required to improve experimental transparency in studies reporting the use of TM techniques for automatic CS in SRs?

RQ2: What information is essential to the reproducibility of TM for CS studies?

RQ3: What information about model complexity should be included in TM based CS studies?

1.3 Original contributions

This thesis reports a novel investigation into the issues surrounding experimental transparency in reporting and how it affects study reproducibility and understanding of the complexity of TM models in automatic CS studies. Specifically, the inadequacies of the current reporting is established, and the most important information to enhance the reproducibility of TM based automatic CS studies identified. More details about how specific units of the work have contributed to knowledge in this area are enumerated below:

i) A mapping study, reported in Chapter 3 is the first in the field to investigate the issue of transparency and to assess the information provided in studies reporting the use of TM techniques to support CS in SR. This work was presented at the 20^{th} International Conference on Evaluation and Assessment in Software Engineering (Olorisade, de Quincey, Brereton, & Andras, 2016).

ii) The reproducibility assessment work reported in Chapter 4 is the first to investigate reproducibility issues in the field. It is also the first to propose a checklist of information that may ensure studies in this area are reproducible. Its findings were reported in the Journal of Biomedical Informatics (Olorisade, Brereton, & Andras, 2017c) and a workshop paper at the 34^{th} International Conference on Machine Learning (Olorisade, Brereton, & Andras, 2017b).

iii) The complexity assessment work reported in Chapter 5 is the first to investigate complexity issues of the TM models for automatic screening of citations in SRs

and the need to report complexity metrics. It is also the first time the Word2vec will be investigated as a feature type in studies for automatic CS. The study findings were presented as a short paper at the 21^{st} International Conference on Evaluation and Assessment in Software Engineering (Olorisade, Brereton, & Andras, 2017a) with an expanded manuscript currently been reviewed for publication in the Research Synthesis Methods journal.

iv) The feature enrichment work reported in Chapter 6 investigates the effect of adding bibliography data to article title and abstract on the performance of models and complexity. The study is the first to compare the performance and complexity of models built from the traditional title and abstract (and optional keywords) with those built by adding bibliography features to the title and abstract. As at the time of writing this thesis, the findings from the study is being prepared for a journal publication.

v) The CS tool - *TeMACS* presented in Chapter 7 is a web based document classification tool which aims to support reviewers in automatic screening of citations in SRs. At the same time, it aims to support automated CS tool researchers by providing information that may help in reproducing its processes and understanding the complexity of its models' decision making. The CS tool is the first of its kind that support explicitly, the transparency and reproducibility of CS in SRs.

Despite the fact that this work is being conducted within a software engineering locale, SRs are used in many disciplines. Research into automated support for CS is most prevalent in the healthcare and software engineering fields. On providing support for the CS stage, more work has been undertaken in the healthcare field. Thus, the challenges highlighted and investigated through studies in this work affects studies from other domains. Therefore, the contributions of this work are not limited to software engineering but can be generalised to other domains.

1.4 Thesis organization

A short description of the chapters that constitute this thesis is presented in this section. A pictorial representation of the relationship between the chapters is presented in Figure 1.2[1].

In **Chapter 2**, a brief theoretical background on ML is provided. Two major learning approaches: the supervised and unsupervised learning were highlighted. The

[1]In the figure, if there are two possible paths exiting a node, different colours are used to indicate the split. The split colours are maintained to highlight the branch path until the main path is rejoined; at which point the initial colour is again used.

chapter introduces TM with a presentation of its process. Performance metrics particularly those relevant to the studies reported in this work are also presented. The background presented in the chapter provides a theoretical context for the work reported in other chapters of the thesis.

In the early stage of the research, a mapping study which evaluates the type and extent of information provided on the TM techniques that are being proposed for the automatic screening of citations in SRs is conducted and presented in **Chapter 3**. The review of the background literature has been presented through a combination of SR methodology and supplementary literature search to ensure up to date information is provided. The review establishes the need for more experimental transparency in the reviewed articles and the potential to investigate support for reproducibility and model complexity issues in candidate articles.

Driven by the outcomes of the review in Chapter 3, **Chapter 4** presents a reproducibility assessment. This study is aimed to assess the reproducibility of selected studies, which are intended to provide automated support for the CS stage of SR. The studies cover both software engineering and medical fields. The assessment is based strictly on the information provided in the selected studies. The work identifies a set of information items that can improve transparency of studies report and its reproducibility. A checklist of these information items is proposed to guide researchers and academic review process.

In continuation of building on the outcomes of the review in Chapter 3, **Chapter 5** presents a study aimed at investigating model complexity and statistical validity issues in TM models to support CS in SR from selected studies. The study builds Support Vector Machine (SVM) models representative of typical models in the selected studies and explore their complexity through the number of Support Vector (SV)s used by the models. The complexity is used to determine whether there is enough concern to warrant its being reported beside being in compliance with scientific requirements. The conclusion indicated high complexity in the models.

Chapter 6 presents a study aimed at investigating how the improvement in the quality of the data with bibliography information will affect model performance and complexity. The study replicates the classification approach in the previous chapter, changing only the data content and the χ^2's α value for reducing the dimension of the feature vector. The conclusions of this study show a strong promise at reducing complexity and increasing performance but its not definitive.

The studies reported in Chapters 4 and 5 show possible effects of the absence of critical information in TM based automatic CS study reports. Whilst maintaining transparency vis-á-vis this information in compliance with the scientific requirement for experimental transparency is useful, effectively communicating a computational study to the extent of being reproducible is challenging. In **Chapter 7**, a novel transparent tool - *"Text Mining based tool for Automatic Citation Screening (TeMACS)"* -

for CS was introduced. *TeMACS* is a document classification is an open web-based tool which incorporates some of the methods and outcomes of studies from previous chapters to demonstrate how a CS tool can be developed to be useful for both SR practitioners and automatic CS researchers. The design, development and features of the tool are reported.

In **Chapter 8**, the findings from the different studies reported in this thesis are brought together and discussed in relation to the original research questions.

In **Chapter 9**, the summary and conclusions from the research undertaken are presented. Recommendations on the use TM based tools and reporting of their corresponding experiments for automatic CS and suggestions for future work are provided.

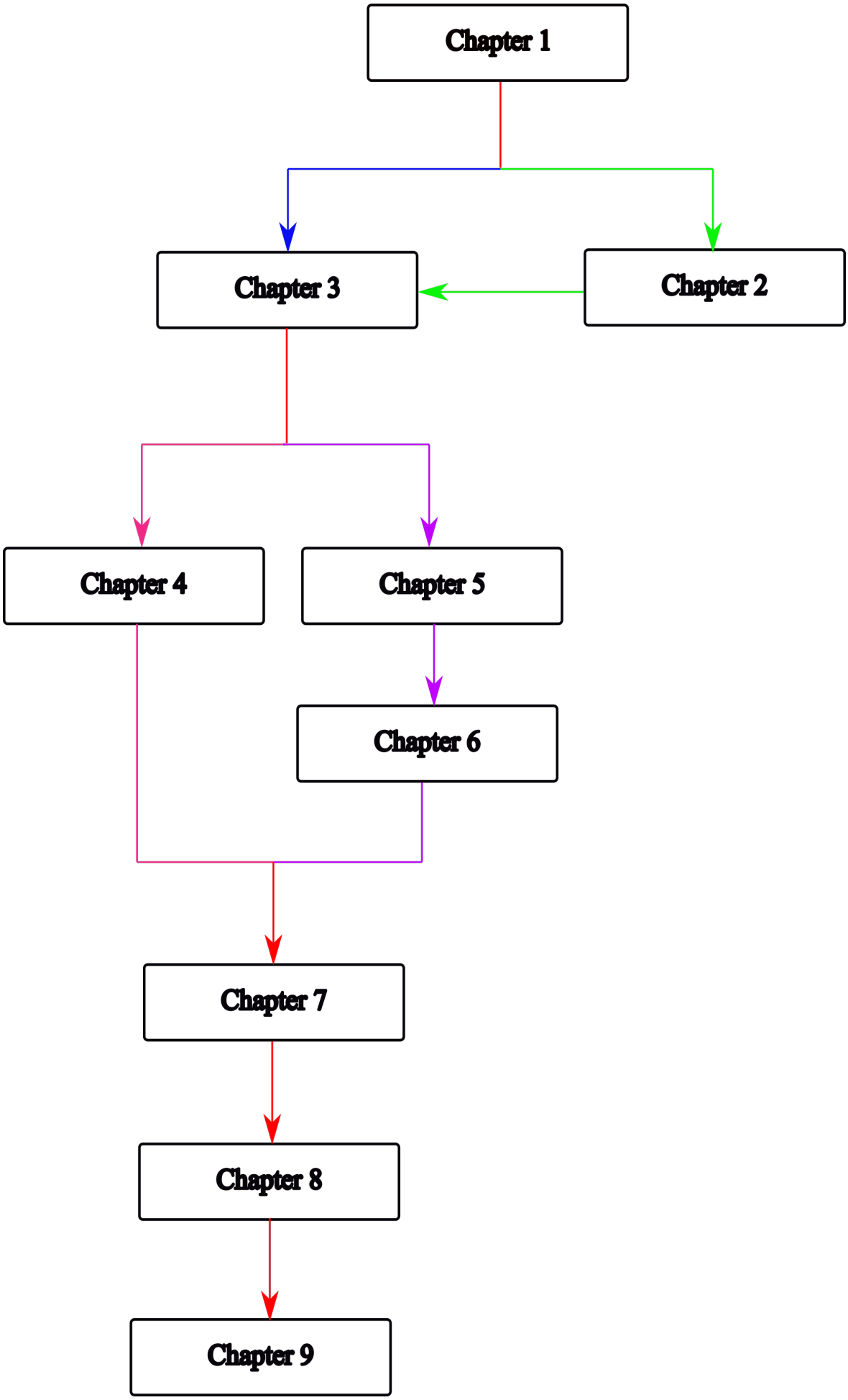

Figure 1.2: Thesis organization

CHAPTER 2

Theoretical Preliminaries

A brief overview of the basics of ML and TM being the core of the subject of the studies in this work is presented in this chapter. The chapter presents a quick introduction to the supervised, semi-supervised and unsupervised learning approaches. The different processes involved in the conduct of TM experiments are also explained. The chapter was rounded with a discussion on model assessment and methods to improve model performance. These concepts underlie the theory behind the work in this research.

In Section 1.1.4, the intensive amount of time and effort consumed by SR was presented. The application of ML algorithms is being explored to support decision making either of the whole SR process or its individual stages. The CS stage has recorded the most success in terms of SR support research with the application of TM techniques. All the studies evaluated in this research have employed one ML algorithm or another. The subject of ML and TM are not native to SR research, therefore, the decision to present a brief overview of the concepts involved to introduce some of the techniques that are relevant to this research.

Many research areas have continued to witness the application of ML techniques to aiding their processes and methods. The case is similar with the application of TM techniques to automatically screen citations during the conduct of SRs. For this purpose, the use of TM related techniques is one of the approaches being explored. TM involves the exploration of textual documents with the aid of analysis tools and technologies to extract useful information. TM supports the "analysis of text with machine using techniques from ML, Information Retrieval (IR), Information Extraction (IE), connecting them with the algorithms and methods of Knowledge Discovery in Databases (KDD), Data Mining (DM), and statistics" (Hotho, Nürnberger, & PaaSS, 2005). TM is like DM but unlike DM, the artefact explored for interesting patterns is not formalized database records but semi-structured or unstructured textual data in documents. The main logic behind the technologies used in TM is that text is turned into some form of structured numerical representation so that ML algorithms can be

applied to large document databases.

Section 2.1 will present a high level overview of machine learning with pointers to where further information can be sourced. This will be followed by a similar introduction to TM in Section 2.2. An introduction to some model assessment metrics particularly relevant to this project is presented in Section 2.3 followed by performance improvement discussions in Section 2.4. The chapter concluded with a summary in Section 2.5.

2.1 Machine learning overview

Learning can be in or through many forms. It can be through the acquisition of new knowledge or cognitive skills, effective representation of new knowledge or new fact discovery through observation and experimentation (Carbonell, Michalski, & Mitchell, 1983; Michalski, Carbonell, & Mitchell, 2013). An indication that learning has taken place is the ability to remember, adapt and generalise to similar situations at a future instance (Marsland, 2015). Given this premise, ML can thus be described as being concerned with learning from data by machines basing their future decisions on previous encounters of similar situations (Marsland, 2015; Murphy, 2012).

There are three major approaches to ML, supervised, unsupervised and reinforcement learnings. Neither this project nor any of the studies reviewed in this thesis touched on reinforcement learning, therefore, only the supervised and unsupervised approaches will be described further in the following sections.

Reinforcement learning has found more use in dynamically interactive environments (Hafner & Riedmiller, 2011; Kaelbling, Littman, & Moore, 1996; Kober, Bagnell, & Peters, 2013). It involves mapping of input to a set of output like the supervised learning (as will be presented in Section 2.1.1) but unlike supervised learning it is not aided by a list of output to learn from since in its case there is often too many possibilities than could be exhausted and often not known ahead. Also, unlike the unsupervised learning (in Section 2.1.2) it does not learn the underlying distribution of the data. The task being addressed by studies contained in this work is a basic binary classification problem which, following from the explanation presented, may have been considered relatively trivial to apply a reinforcement learning algorithm. Also, in reinforcement learning, there is a delayed feedback indicating the goodness of a series of decisions, however this scenario does not fit the SR/ CS context where decisions on inclusion/exclusion criteria are independent and based in principle on a priori set criteria; reinforcement learning could therefore be considered applicable if the inclusion/exclusion criteria would not be pre-set and the aim would be to learn these as well. However, this is not appropriate in the context of SR. These reasons (and may be more) might account for why the reinforcement learning approach was

not found used in any of the work evaluated.

A possibly fourth type of learning approach is referred to as semi-supervised learning. Semi-supervised learning is a combination of both the supervised and unsupervised learning. The model is trained with a limited number of input-output mappings and it use the knowledge to project the output class of the rest of the data (Zhu, 2006; Hady & Schwenker, 2013).

2.1.1 Supervised learning

Supervised learning is the process of creating a classifier that learns a set of rules from provided instances to generalize to new ones (S. B. Kotsiantis, Zaharakis, & Pintelas, 2007). In the process, a general inductive process, called the learner is fed with some training set of documents $\mathcal{D}$ that have been labelled according to their pre-defined classes $\mathcal{C}$. The goal is to learn a mapping from input $\mathfrak{d}$ to target output $\mathfrak{c}$ given the input-output pairs (Equation 2.1) (Murphy, 2012).

$$\Phi = \{(\mathfrak{d}_i, \mathfrak{c}_j)\}_{i=j}^{N}; \mathcal{D}X\mathcal{C} \rightarrow \{T, F\} \tag{2.1}$$

The learner will generate a model (classifier) based on its observed characterisation of the constituents of the different categories. Then, the classifier can be used to determine the class of previously unseen documents based on the provided categories. In its simplest setting, each training input $\mathfrak{d}_i$ is a n-dimensional vector of number representations of the features. When $\mathfrak{c}_j$ is categorical, the learning problem is called *classification* and it is called *regression* when the target output is real-valued.

The Support Vector Machine (SVM) is an example of a supervised ML technique and as will be shown in Section 3.4.2, it is the most used algorithm among the studies on automatic CS. A SVM is a supervised learning technique applicable to both classification and regression. It is based on the structural risk minimization theory introduced by Cortes and Vapnik (1995). In its simplest (linear) form, it is a binary classification model that seeks an optimal separation margin (optimal separating hyperplane) between the positive and negative examples (see Figure 2.1), where margin refers to the minimal distance from the separating hyperplane to the closest data points (Hearst, Dumais, Osuna, Platt, & Scholkopf, 1998; S. B. Kotsiantis et al., 2007; Murphy, 2012).

Given a training set of input-output pair $(x_i, y_i), i = 1, \cdots, l$ where $x_i \in R^n$ and $y_i \in \{1, -1\}$, the SVM requires the solution of the optimisation problem in Equation 2.2 (Cortes & Vapnik, 1995).

$$\min_{\mathrm{w},b,\xi} \frac{1}{2}\mathrm{w}^T\mathrm{w} + C\sum_{i=1}^{l}\xi_i \tag{2.2}$$

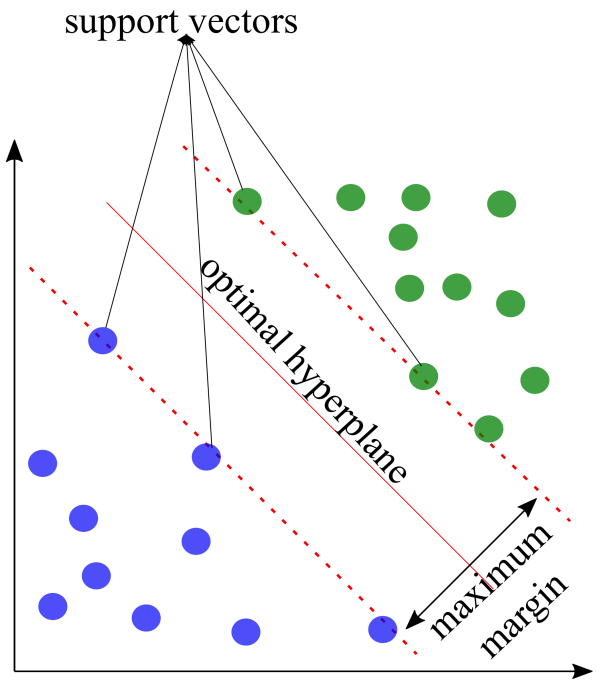

Figure 2.1: Illustration of SVM classification for linearly separable data

subject to $(y_i \mathrm{w}^T x_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0$

To classify non-linear data (Figure 2.2), the SVM transforms the input vector into a very high dimensional feature space using non-linear transformation functions (called the "kernel trick"), where the data points can be separated linearly (Figure 2.3). In the non-linear case, the solution in Equation 2.2 is solved subject to $(y_i \mathrm{w}^T \phi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0$. Examples of some popular kernels: linear, polynomial, radial basis function (rbf) and sigmoid are presented below:

$$K(X_i \cdot X_j) = \begin{Bmatrix} X_i \cdot X_j & \text{linear} \\ \gamma X_i \cdot X_j + C & \text{polynomial} \\ \exp(-\gamma |X_i \cdot X_j|^2) & \text{rbf} \\ \tanh(\gamma X_i \cdot X_j + C) & \text{sigmoid} \end{Bmatrix}$$

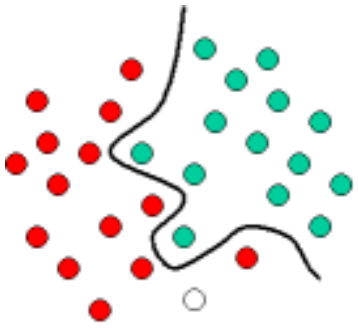

Figure 2.2: Illustration of SVM classification of non-linearly separable data[1]

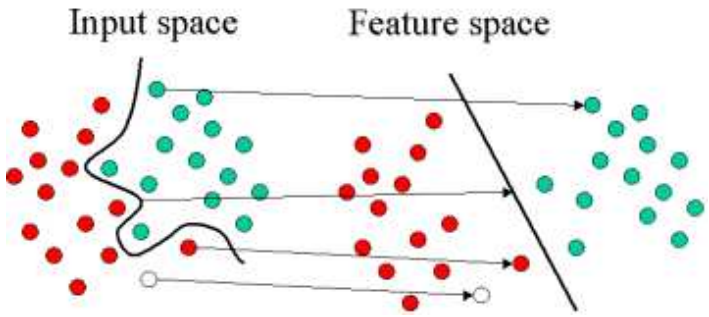

Figure 2.3: SVM projection of non-linear data with kernel trick

The SVM's decision is based only on data points closest to the margin called Support Vector (SV)s (A. Khan, Baharudin, Lee, & Khan, 2010; S. B. Kotsiantis et al., 2007). The SVM have some other parameters apart from the kernel whose values must be tuned to obtain optimal performance from the algorithm. Key of these parameters are the 'C' and gamma parameters.

'C' is a regularisation parameter which maintains the trade-off between achieving a low error on the training data and minimising the norm of the weights - better generalisation. As the value of 'C' increases the complexity of the model increases which may lead the model to over-fitting its data. Given the objective function in Equation 2.2, if 'C' is too large, the optimisation algorithm will try to reduce $|w|$ as much as possible leading to a hyperplane which tries to classify each training data correctly (Alvarsson et al., 2014). This process will lead to the algorithm overfitting. On the other hand, if the value of 'C' is too small, the objective function will take the affinity to increase it a lot which will result in a large training error and by implication, underfitting (Cherkassky & Ma, 2004a; Alvarsson et al., 2014).

The gamma parameter defines the extent of the reach of influence of a single training data where 'low' values signifies 'far' and 'high' value signifies 'close'. Gamma can be seen as the inverse of the radius of the data samples selected as SVs (Alvarsson et al., 2014). This means if the radius is too large, the region of influence of the SV only includes the SV itself and regularisation with 'C' cannot prevent overfitting that results. When the gamma is very small, the region of influence of any selected SV covers the whole training data (Cherkassky & Ma, 2004a).

Following from the above discussion, it is clear that setting 'C' and gamma to optimal values is key to the performance of the SVM algorithm. Some of the studies reported in other parts of this thesis (Sections 4.2.1.4,5.4.6 and 6.3.2) explore the use of SVM models in text classification.

[1]Source: http://www.statsoft.com/Textbook/Support-Vector-Machines

2.1.2 Unsupervised learning

Clustering is a prime example of unsupervised learning. In unsupervised learning, the learner is provided with no predefined classes ($\mathcal{C}$), only the input documents ($\mathcal{D}$). The goal is for the learner to explore characteristics of the instances and discover "interesting patterns" that it will use to partition the instances into a finite number of clusters ($\mathcal{K}$) ensuring that members of a cluster share more similarities than those of other clusters in the data (Fahad et al., 2014; Verma, Srivastava, Chack, Diswar, & Gupta, 2012). According to Murphy (2012), this usually involves two steps. The first step involves the estimation of the distribution over the number of clusters, $p(\mathcal{K}|\mathcal{D})$ which is approximated as shown in Equation 2.3.

$$\mathcal{K}^* = \underset{\mathcal{K}}{\operatorname{argmax}}\, p(\mathcal{K}|\mathcal{D}) \tag{2.3}$$

The second step involves the estimation of the cluster each data point i belongs to as shown in Equation 2.4.

$$z_i^* = \underset{k}{\operatorname{argmax}}\, p(z_i = k|\mathfrak{d}_i, \mathcal{D}) \tag{2.4}$$

where, z_i is a latent variable explored by the model and $z_i \in \{1, \ldots, \mathcal{K}\}$ denotes the cluster assigned to data point i. There are several types of clustering techniques and algorithms, the main ones and their examples are reviewed and explained in (Fahad et al., 2014; Murphy, 2012).

2.2 Text mining: an introduction

The goal of TM is to exploit vast amounts of information from multiple documents and sources by using automated means to categorise and characterise them into a fixed number of (pre-defined) categories, where each document $\mathfrak{d}$ can be in none, one or more categories (Inzalkar & Sharma, 2015; Joachims, 1998; K. Sharma, Sharma, Joshi, Vyas, & Bapna, 2017). The discussion on TM in this thesis will focus more on text classification. The text classification process involves text retrieval, preprocessing, dimensionality reduction, model training (or development) and assessment. These steps are depicted in Figure 2.4 and discussed in more detail in the following sections.

2.2.1 Data retrieval

In this age of big data, the data required for a TM task are often located in some remote locations or may need to be retrieved from multiple sources e.g. websites. Therefore, it is important to capture the source(s), nature and portion of the data and the method used to retrieve them. The data may be stored in a database which will

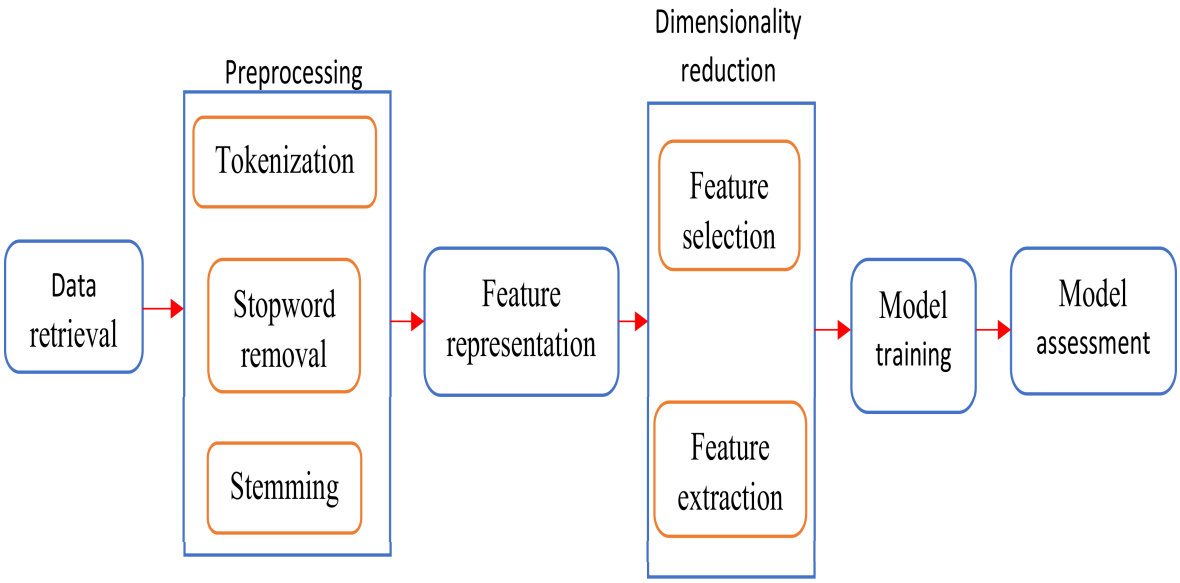

Figure 2.4: Text mining process

then require a retrieval method as simple as a Structured Query Language (SQL) or located in multiple websites where the retrieval method will involve writing scripts to automatically scrape the websites and extract targeted information from them. The dataset may also be stored in an original format (e.g. json or eXtended Markup Language (XML)) that may require writing of scripts to reformat and/or extract portions of interest. In essence, the description of the data structure, storage and retrieval method is always a good practice and essential to the TM process.

2.2.2 Preprocessing

Texts cannot be handled by the ML algorithms, thus they need to be converted into a numerical format that can be processed and managed by algorithms. After retrieval, the different words in all the documents of interest are separated into individual words (tokenization) and collated in a single entity, a feature vector called Bag-of-Words (BOW) (Lebanon, Mao, & Dillon, 2007). The BOW approach did not take the semantic context of each word into consideration. Other Natural Language Processing (NLP) approaches utilising the semantic context of each word exist but will not be explored in this report. The preprocessing step involves tokenization, stopwords removal and stemming or any other chosen NLP activities. The advantage of undertaking preprocessing in text classification was reported in (Uysal & Gunal, 2014). These tasks are further discussed in the following sections.

2.2.2.1 Tokenization

Tokenization refers to the process of cleaning, extracting and separating individual (unique) words in all the documents and storing them (usually) as independent

terms (Miner, Elder IV, & Hill, 2012).

2.2.2.2 Stopwords removal

Another task sometimes undertaken in pre-processing is the removal of commonly appearing 'non-informative, non-content' words in categories of prepositions, auxiliary verbs, articles, conjunctions, special characters and numbers. An example of these stopwords is given in (Fox, 1989). These words are considered to convey no particular meaning or are not useful to discriminate between documents. The effect of removing them has been evaluated (Srividhya & Anitha, 2010). This has become a common practice in TM studies.

2.2.2.3 Stemming

Stemming is a language normalization step in text preprocessing (Miner et al., 2012). A stemming algorithm removes inflection (suffix, prefix, or any other transformation) from the different words and reduces each to its original root. This is to create uniformity among similar words and reduce unnecessary duplicity. For example, words like "going and gone" will both be reduced to "go". It should be noted, that despite being useful, this method has a downside of merging two or more words with the same sound (homonyms) and spelling (homographs) but different meanings and root as being the same e.g. *book* (noun), *book* (verb) and *booking* (present participle of verb book), *bookings* (noun) four words of three categorical meaning will all be reduced to "book". The most widely used stemming algorithm is the "porter stemming algorithm" (Porter, 1980).

2.2.3 Feature representation

After the text has been preprocessed, it is then encoded or weighted in numerical form and stored in a data structure (feature vector) ready for the learning algorithm. This step is called feature representation. The feature vector relies on the Vector Space Model (VSM), an algebraic model for representing text documents. In VSM, each document is mapped against the words that it contains using frequency based schemes like term frequency (tf) or term frequency-inverse document frequency (tf-idf). Given a document $\mathcal{D}$, it can be represented with a vector as expressed in Equation 2.5,

$$\mathcal{D} = (\mathfrak{d}_{i1}, \mathfrak{d}_{12}, \cdots, \mathfrak{d}_{in});\ \mathfrak{d}_{ij} \implies weight\ of\ the\ j_th\ term \tag{2.5}$$

These representations are combined in a high dimensional term-document matrix know as the Bag-of-Words (BOW). The BOW is the most commonly used vector to represent a corpus prior to classification or clustering. The BOW is an orderless representation of a document as the multi-set of its constituent words without regard for

grammar but reflecting the importance of the word to the document (Korde & Mahender, 2012). Some of the most commonly used feature representation techniques (binary, term frequency (tf) and the term frequency-inverse document frequency (tf-idf)) are briefly introduced below. For more details on feature representation or term weighting and the different approaches and types, see (Ikonomakis, Kotsiantis, & Tampakas, 2005; Sebastiani, 2002; Leopold & Kindermann, 2002).

2.2.3.1 Binary feature

The presence of a feature f in a document is denoted by a '1' and its absence signified by a '0' in the term-document matrix irrespective of the number of times it occurred. Given a document d, the binary representation of a feature f can be expressed as in Equation 2.6 below:

$$h(f) = \begin{cases} 1, & \text{if } f \in d \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

2.2.3.2 Term frequency

This is the representation of a term in a document by the number of occurrences of such terms in the document. This can be expressed as shown in Equation 2.7:

$$tf(f_i, d_j) = \frac{freq_{ij}}{max_k freq_{kj}} \tag{2.7}$$

2.2.3.3 Term frequency-inverse document frequency

The frequency representation is sometimes normalized and one of the often used count normalization techniques to represent features is the tf-idf. tf-idf is expressed as the relative frequency of a term or feature f in a specific document d normalised by the inverse proportion of the feature over the entire document D product of the tf and the feature's inverse document frequency. The inverse document frequency (idf) is obtained by taking the logarithm of the corpus size divided by the number of documents containing the word. Given a document collection D, a feature f, and an individual document $d \in D$, the tf-idf of f relative to d can be calculated as shown in Equation 2.8 (Robertson, 2004; Salton & Buckley, 1988):

$$f_d = freq_{f,d} * \log(\frac{|D|}{freq_{f,d}}) \tag{2.8}$$

2.2.3.4 Word2vec

The word2vec is a predictive model for learning word embedding from raw text proposed by Mikolov, Sutskever, Chen, Corrado, and Dean (2013). The model works

by first creating a vocabulary from the training text data and then learning the vector representation of words incorporating an understanding of when and how often words are used together in the representation (Mikolov, Sutskever, et al., 2013; Mikolov, Chen, Corrado, & Dean, 2013). It learns by creating a shallow neural network architecture called the *skip-gram model*. The *skip-gram model* consists of an input layer, a projection layer and an output layer to learn and predict nearby features. The individual feature vector is trained to maximize the log probability of neighbouring features in a corpus as expressed in Equation 2.9 (Kusner, Sun, Kolkin, & Weinberger, 2015; Mikolov, Chen, et al., 2013), i.e., given a sequence of features $f_1, \cdots, f_T$,

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{j\in nb(t)}\log p(f_j|f_t) \tag{2.9}$$

where $nb(t)$ is the set of neighbouring features of feature f_t and $p(f_j|f_t)$ is hierarchical softmax of the associated feature vectors v_{fj} and v_{ft} (see (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013) for more details). Average words word2vec is used to represent features in the study reported in Chapter 5 (Section 5.4) and Chapter 6 (Section 6.3). The average word2vec incorporates the average of each word over the given corpus.

2.2.4 Dimensionality reduction

The dimension of the feature vector is the number of documents in a corpus by the number of unique terms in the corpus. It is not uncommon in text classification for the size of this vector to run into orders of tens of thousands; a size of this magnitude usually affects the performance, accuracy and processing power requirements of the text classifier. This phenomenon is referred to as the *curse of dimensionality*. Therefore, it is common practice to first reduce the dimension of the vector. Dimensionality reduction involves the application of statistical manipulation activities to reduce the vector size to a minimally manageable dimension by determining the set of terms considered to be most descriptive of each document and at the same time discriminative of others. The reduction process is achieved through feature selection and feature extraction. These two approaches are further discussed below.

2.2.4.1 Feature selection

Feature selection is a term commonly used in data mining to describe the tools and techniques available for reducing inputs to a manageable size for processing and analysis. The aim of feature (term) selection techniques is to reduce the size of the feature vector by removing the irrelevant features i.e. create a subset of the original feature space with only the features deemed to have strongest predictive

power (Ikonomakis et al., 2005; Refaeilzadeh, Tang, & Liu, 2009). This is achieved by setting arbitrarily, the number of top features to select e.g. top 5%, ranked according to scores from the methods. According to Sebastiani (2002), the process seeks a set τ from the original set $\mathcal{T}$ of terms ($|\tau| \ll |\mathcal{T}|$) that results in the best performance when used for term weighting. The most common feature selection algorithms are: tf, Chi-square (χ^2) statistic, information gain, and mutual information. These methods are discussed to the point and compared in (Forman, 2003; Y. Yang & Pedersen, 1997).

i) The χ^2 statistic can be expressed as in Equation 2.10:

$$\chi^2(\mathfrak{f}, \mathfrak{c}) = \frac{\mathcal{N} \times (\mathcal{A}\mathcal{D} - \mathcal{C}\mathcal{B})^2}{(\mathcal{A} + \mathcal{C}) \times (\mathcal{B} + \mathcal{D}) \times (\mathcal{A} + \mathcal{B}) \times (\mathcal{C} + \mathcal{D})} \tag{2.10}$$

based on a two-way contingency table of feature $\mathfrak{f}$ and class $\mathfrak{c}$ where,
$\mathcal{A}$ is the number of co-occurrence instances of $\mathfrak{f}$ and $\mathfrak{c}$,
$\mathcal{B}$ is the number of times $\mathfrak{f}$ occurs without $\mathfrak{c}$,
$\mathcal{C}$ is the number of times $\mathfrak{c}$ occurs without $\mathfrak{f}$,
$\mathcal{D}$ is the number of times neither $\mathfrak{f}$ nor $\mathfrak{c}$ co-occur, and
$\mathcal{N}$ is the total number of documents (Y. Yang & Pedersen, 1997). In feature selection, the χ^2 statistic is used to rank features in order of importance and not used to make statements about dependence or independence of variables. Therefore, the α value as used in TM feature selection corresponds to the value of the top percentile of the ranked features to retain as training data. An alternative choice found in the χ^2 implementation for feature selection is the actual number of top features to retain as against percentile. Throughout this research, α value refers to the top percentile of features to retain based on the result of the χ^2 feature ranking technique.

ii) Mutual information is a measure used in statistical language modelling of words associations and related applications (Y. Yang & Pedersen, 1997). Mutual information is mathematically expressed in Equation 2.11:

$$\mathcal{I}(\mathfrak{f}, \mathfrak{c}) = \log \frac{P_r(\mathfrak{f} \wedge \mathfrak{c})}{P_r(\mathfrak{f}) \times P_r(\mathfrak{c})} \tag{2.11}$$

which is estimated with,

$$\mathcal{I}(\mathfrak{f}, \mathfrak{c}) \approx \log \frac{\mathcal{A} \times \mathcal{N}}{(\mathcal{A} + \mathcal{C}) \times (\mathcal{A} + \mathcal{B})}$$

In the information-theoretic sense, Mutual information measures how much information a feature contains about a class. A measure of 0 indicates that the

distribution of a feature given a class is no better than the feature's distribution in the whole document while the measure reaches its maximum if the feature is a perfect indicator of the class.

iii) Information gain is used in ML as a measure of feature goodness. It utilised the knowledge of the presence or absence of a feature in a document to measure the amount of bits obtained for classification prediction. When a feature is considered in isolation, the information gain results to a 0 value but when considered with other features a predictive values is produced. The closer to 1 this value is, the better the feature is at helping a model make a good prediction based on the classes.
Given a target output classes $\{c_i\}_{i=1}^{m}$, the information gain can be expressed generally as shown in Equation 2.12:

$$\begin{aligned}\mathcal{IG}(\mathfrak{f}) = & -\sum_{i=1}^{m} P_r(\mathfrak{c}_\mathfrak{i}) \log P_r(\mathfrak{c}_\mathfrak{i}) \\ & + P_r(\mathfrak{f}) \sum_{i=1}^{m} P_r(\mathfrak{c}_\mathfrak{i}|\mathfrak{f}) \log P_r(\mathfrak{c}_\mathfrak{i}|\mathfrak{f}) \\ & + P_r(\bar{\mathfrak{f}}) \sum_{i=1}^{m} P_r(\mathfrak{c}_\mathfrak{i}|\bar{\mathfrak{f}}) \log P_r(\mathfrak{c}_\mathfrak{i}|\bar{\mathfrak{f}})\end{aligned} \tag{2.12}$$

where,
P_r implies the probability of its expression, and $\bar{\mathfrak{f}}$ implies the situation where a feature was not found.

The mutual information, information gain and the χ^2 algorithms for feature selection belong to the class of evaluation criteria used by a feature selection approach called the 'filter method' (Jain & Singh, 2018). In the filter method for feature selection, the selection is done independent of the implementation of any learning algorithm. The method rank features based on certain evaluation criteria e.g. mutual information, χ^2 and many more (J. Tang, Alelyani, & Liu, 2014; Aggarwal, 2014). The evaluation algorithms for the filter approach are deemed fast and efficient and are so preferred on voluminous data. However, because they are executed independent of the learning algorithm, they tend to miss interaction among classifiers and dependency of one feature over another and may lead to their failure to chose the most 'useful' features (Ang, Mirzal, Haron, & Hamed, 2016).

2.2.4.2 Feature extraction

Similar to feature selection, feature extraction techniques are also used to reduce the size of feature vectors but unlike feature selection, it does not perform ranking nor use any weighting method but creates a more compact (new) feature set rather than

removing the low-information features (Miner et al., 2012). The Principal Component Analysis (PCA) and Latent Semantic Indexing (LSI) are two of the well-known functions for performing feature extraction (Meng, Lin, & Yu, 2011; Uuz, 2011). These two are briefly introduced below:

i) The PCA is a widely used dimensionality reduction method in ML. It provides a guideline for how to reduce a complex high dimensional dataset to one of lower dimensionality to reveal any hidden, simplified structures that may underlie it (Jolliffe, 2002; Abdi & Williams, 2010; Murphy, 2012). PCA determines the eigenvectors and eigenvalues of a matrix to project a dataset to a new coordinate system (Jolliffe, 2002). The projection involves calculation of a covariance matrix of a dataset to minimize the redundancy and maximize the variance (Murphy, 2012; Fodor, 2002; Abdi & Williams, 2010). Given a dataset, the covariance of X and Y is given as in Equation 2.13:

$$cov(X,Y) = \sum_{i=1}^{N} \frac{(x_i - \bar{x})(y_i - \bar{y})}{N} \tag{2.13}$$

where, $\bar{x}$ and $\bar{y}$ are means of X and Y respectively and N, the dimension of the dataset. The covariance matrix is a matrix A with elements $A_{i,j} = cov(i,j)$. In order to map a high-dimensional dataset to a lower dimensional space, the best low-dimensional space that minimizes the error between the dataset and the PCA is determined by the eigenvectors of the covariance matrix using the criterion in Equation 2.14:

$$\frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{N} \lambda_i} > \theta \tag{2.14}$$

where, θ is a predetermined threshold value, K is the selected dimension from the original matrix of dimension N and λ is an eigenvalue (Fodor, 2002). A common method for finding the eigenvalues and eigenvectors is the Singular Value Decomposition (SVD). SVD is based on a linear algebra theorem which states that a rectangular matrix A can be broken into the product of three matrices:

a) an orthogonal matrix U,

b) a diagonal matrix S, and

c) the transpose of an orthogonal matrix V.

The general approach for computing SVD is:

$$A = USV^T \tag{2.15}$$

where,

$$A \in R^{(m \times n)}\ with\ (m \geq n),$$

$$U \in R^{(m \times m)},$$

$$V \in R^{(n \times n)},$$

and S is a diagonal matrix of size $R^{(m \times n)}$ and $U^T U = I$, $V^T V = I$; the columns of U are orthogonal eigenvectors of AA^T, the columns of V are orthonormal eigenvectors of $A^T A$, and S is a diagonal matrix containing the square roots of eigenvalues from U or V in descending order (Murphy, 2012; Kumar & Chandrasekhar, 2012).

ii) LSI is another method that relies on SVD but operates on the term-document matrix. It provides dimensions with semantic relation where features in the same dimension are usually topically related (Miner et al., 2012; Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990; Dumais, Furnas, Landauer, Deerwester, & Harshman, 1988). LSI is a mathematical approach using SVD to estimate the latent semantic structure that is possibly obscured by variability in word usage. It extracts and represents the contextual meaning of words based on usage by statistical computations applied to a large corpus of text. The logic behind LSI is that the context in which words appear or not offer a set of mutual constraints that largely determines the similarity of meaning of words and sets of words to each other. The LSI relies on transformation like the PCA, so the results are not just simple contiguity frequencies, co-occurrence counts, or correlations in usage, but depend on a powerful mathematical analysis that is capable of correctly inferring much deeper (latent) relations between features. Though, the LSI makes no use of word order or morphology, it still manages to extract correct reflections of passage and word meanings quite well.

2.2.5 Model training

After the feature selection or feature extraction process, the next step is to train one or more classifiers using any ML algorithm of choice. A typical training during text classification commence with the division of the dataset into three (or two) portions. One portion usually larger than the rest is used to train the model, a second optional portion is used for intermediate assessment (validation) of the model while a third hold out portion is used for final assessment of the trained model (S. B. Kotsiantis, Zaharakis, & Pintelas, 2006). Where the data size is relatively small, it is divided into only two portions.

The choice of the classification algorithm to use is critical and always not definite from inception of the model training process. Therefore, a common practice is to undergo a model selection phase.

Table 2.1: Confusion matrix

actual / predicted	Positive	Negative
positive	true positive (TP)	false negative (FP)
negative	false positive (FN)	true negative (TN)

In model selection, since the underlying relationship pattern among features is unknown, the idea is to let the data chose the model that best represents it. The practice is to start out with several learning algorithms or same algorithm run against a grid search of different parameter setting (Browne, 2000); each possible combination of parameters is then trained on the training set, after the training, the trained model is tested on the validation set after which the best performing one(s) is/are chosen to be tested on the hold out set (discussion on model selection is continued in Section 5.2). This process often involves combining the grid search with Cross Validation (CV) (more discussion on CV is presented in Section 2.4).

2.3 Model assessment

Following a successful training of a model of choice, it is tested for how it can perform generally on input it has never encountered. The test portion (or hold out set) of the input is used for this exercise. The input is fed into the model and the model output is compared to the expected output based on pre-knowledge of the document classes. The most basic metrics used to estimate classifier performance from which all others are mostly derived are:

i) True Positive (TP): This is expressed as the of the count of positive class (relevant) documents that are so classified by a model.

ii) True Negative (TN): The number of negative class (irrelevant) documents correctly classified as negative by a model.

iii) False Positive (FP): The number of negative documents that were wrongly classified as positive documents by a model.

iv) False Negative (FN): The number of positive documents misclassified as negative

These four measures are often presented in a grid form called the confusion matrix as shown in Table 2.1. There are numerous other metrics for assessing the performance of the TM model, a few of them relevant to the studies presented in this thesis are presented as follows - recall (Equation 2.16), precision (Equation 2.17), accuracy (Equation 2.18), F1 (Equation 2.19), Work Saved over Sampling (WSS) (Equation 2.20) and Matthews Correlation Coefficient (MCC) (Equation 2.22).

i) Recall: Recall is the fraction of correctly classified positive examples by the total positive examples in the whole corpus.

$$recall = \frac{TP}{TP + FN} \tag{2.16}$$

ii) Precision: Precision is the ratio of actual positive examples and the total positive prediction.

$$precision = \frac{TP}{TP + FP} \tag{2.17}$$

iii) Accuracy: Accuracy is the fraction of the total correct negative and correct positive prediction by corpus size.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.18}$$

iv) F_1: F_1 score is the weighted harmonic mean of the recall and the precision.

$$F1 = 2.\frac{precision.recall}{precision + recall} \tag{2.19}$$

The closer to 1 the value of these measures - recall, precision, accuracy and the F-measure - the better the performance of the model being measured.

v) WSS: given a certain recall level (rl), the WSS is the percentage of the articles initially returned by the literature search which the researcher would not have to read because they have been screened out by the model (A. M. Cohen, Hersh, Peterson, & Yen, 2006). A WSS score of 0 or less at a particular recall level indicates that the model would not be saving any work in relation to the number of citations to be screened beyond a random choice. This measure is though more suitable to a ranking algorithm (Howard et al., 2016).

$$WSS@rl = \frac{(TN + FN)}{N} - 1 + \frac{TP}{TP + FN} \tag{2.20}$$

based on Equation 2.16, equation 2.20 can be alternatively expressed as:

$$WSS@rl = \frac{(TN + FN)}{N} - 1 + recall \tag{2.21}$$

where *rl* refers to the recall level for which the measure is taken.

vi) MCC: MCC is a measure of the quality of the classifications of a binary classific-

ation model (Matthews, 1975). Its value ranges from:

$$mcc(f) = \begin{cases} +1, & \text{perfect prediction} \\ 0, & \text{prediction not better than a random guess} \\ -1, & \text{strong disagreement between the true data and the prediction} \end{cases}$$

and can be expressed as shown in Equation 2.22.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{2.22}$$

Details on some of these and other model assessment metrics can be found in (Japkowicz & Shah, 2011; Murphy, 2012; Japkowicz, 2013; Menardi & Torelli, 2014). These metrics are used across the studies presented in this thesis (refer to Sections 5.5 and 6.4.4.1).

There is hardly any of the measures without its own weakness or bias. Considering the values in Tables 2.2(a) and 2.2(c), the result of precision and recall for Tables 2.2(a) and 2.2(c) is equal. Though, they both exhibit similar positive recognition potential but the difference in the negative recognition potential (strong to nil) was not reflected in both measures; accuracy will reflect this type of difference. Given the values presented in Tables 2.2(a) and 2.2(b), accuracy will be 60%, however, the performance distribution in both table differ. Table 2.2(a) is weak at detecting positive examples but strong at negative examples while Table 2.2(b) is vice versa. The accuracy measure was in no way reflective of actual performance strength of the classifier behaviour in this case. Accuracy is not robust to classifier performance in a class imbalance situation, given a dataset of about 95% negative example, a classifier that indicates every sample as negative will have accuracy of 95% whereas a classifier that is more critical will have a less accuracy value.
The MCC uses a correlation approach utilising all the four basic metric and may often produce a much more balanced evaluation of the prediction. However, in cases where where there is no or very few FPs with few TPs at the same time, the MCC may be relatively high (Baldi, Brunak, Chauvin, Andersen, & Nielsen, 2000).

2.4 Performance reliability and improvement

In this section, two concepts - CV and ensemble methods are discussed. CV is a practice conducted to capture a model's more reliable performance while the ensemble method is employed to in most cases improve classification results by combining the decision of multiple classifiers on the same data. These two concepts are further introduced in following subsections.

Table 2.2: Confusion matrix to illustrate metrics' pros and cons

(a)

predicted \ actual	Positive	Negative
positive	2000	1000
negative	3000	4000

(b)

predicted \ actual	Positive	Negative
positive	4000	3000
negative	1000	2000

(c)

predicted \ actual	Positive	Negative
positive	2000	1000
negative	3000	0

2.4.1 Cross validation

Some level of assurance about the reliability of the future performance of a trained model is important to practitioners. Therefore, to minimize the model prediction error range and obtain a performance more representative of a model's ability, the process of CV is often employed. In CV, the whole input data is divided into equal sized mutually exclusive subsets in such a way that all samples will at some point act as training data and at other time as test data. Ensuring the learning algorithm's parameters are kept constant, multiple models are developed from different portions of the dataset and tested each time on a different portion. The performance metrics are recorded on each fold's run and the average at the end is taken as the mean performance of the model given the dataset (S. B. Kotsiantis et al., 2006; Schaffer, 1993; Browne, 2000). Following are the possible ways of conducting cross validation:

i) k-fold CV: In a k-fold CV, the entire dataset is divided into k equal partitions (called folds). Then, there will be k run of the training and testing process, on each run, the model will be trained on $k-1$ fold and tested on the k_th fold excluded from training (Refaeilzadeh et al., 2009). Different fold is selected at on each run until all folds are exhausted. Figure 2.3 illustrates this approach using a 5-fold CV. The fact that every data sample will be utilized as a training and test data is seen as an advantage of k-fold CV. Also, the possible error of mis-splitting or biased splitting of the dataset into test and train sets is avoided (Witten, Frank, Hall, & Pal, 2016). k can be any number but the most commonly used number is 10.

ii) Leave one out CV (LOOCV): In LOOCV, the splitting is done according to the

Table 2.3: 5-fold CV illustration

K	fold-1	fold-2	fold-3	fold-4	fold-5
k = 1	test set	train set	train set	train set	train set
k = 2	train set	test set	train set	train set	train set
k = 3	train set	train set	test set	train set	train set
k = 4	train set	train set	train set	test set	train set
k = 5	train set	train set	train set	train set	test set

number of available data samples. Therefore, with N-data samples, N splitting is done and $N-1$ samples are used for training while the nth sample is used to test. This is an extreme case of the k-fold CV (Refaeilzadeh et al., 2009).

iii) $n \times k$ fold CV: Another popular CV approach is the $n \times k$-fold CV, where k is the number of folds (as defined for k-fold CV above) and n is the number of repeated runs. The most popular of this approach is the 5×2-fold CV.

A more general discussion on CV can be found in (Browne, 2000; Refaeilzadeh et al., 2009).

The 5×2 fold CV was used in the study reported in Section 4.2.1.4, the approach was combined with the 2×5-fold CV in the studies reported in Sections 5.4.6 and 6.3.

2.4.2 Ensemble learning

An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some ways (usually by weighted or unweighted voting) to classify new examples (Dietterich, 2002; Rokach, 2005). The objective of using ensemble of classifiers is to accomplish better accuracy on the training set and a better generalisation over unseen data (P. Yang, Hwa Yang, B Zhou, & Y Zomaya, 2010; S. Wang et al., 2009). Methods of constructing the best ensembles of classifiers is still an active research area; some of the current general purpose approaches applicable to many different algorithms are:

i) Bagging: Bootstrap Aggregation is a special case of model averaging approach; it trains the classifier with different subset of the training examples, drawn randomly with replacement over a certain number of times bootstrap replication. On each run, a different subset of the training example is used to train a different classifier of the same type (Dietterich, 2002). The classifiers are then combined through a majority decision. It has been pointed out in studies that this method can achieve better generalisation by reducing variance (Breiman, 1998).

ii) Boosting: Like the bagging method, boosting also creates an ensemble of classifiers through repeated sampling of the training example. A major difference is

that boosting strives to provide the most informative training dataset for each consecutive classifier (Opitz & Maclin, 1999; Freund & Schapire, 1996). Each iteration of boosting creates three weak classifiers:

a) The first classifier $C1$ is trained with a random subset of the available training data.

b) The training data subset for the second classifier $C2$ is chosen as the most informative subset, given $C1$. Specifically, $C2$ is trained on a training data only half of which is correctly classified by $C1$, and the other half is misclassified.

c) The third classifier $C3$ is trained with instances on which $C1$ and $C2$ disagree.

The three classifiers are combined through a three-way majority vote.

iii) AdaBoost: Adaptive Boosting is a specific type of the Boosting approach. It manipulates the training examples to generate multiple hypotheses. It maintains a set of weights (w) over the training examples which are adjusted on each iteration l by invoking the learning algorithm to minimize the weighted error on the training set and returns a hypothesis h_l. The weighted error of h_l is calculated and applied to update the weights on the training examples (Freund & Schapire, 1996). The essence of the weight changes is to assign more weight to the misclassified training examples and less to those correctly classified by h_l. The final classifier

$$h(x) = \sum_{l=1}^{L} w_l h_l(x) \tag{2.23}$$

is constructed by a weighted vote of the individual classifiers (Dietterich, 2002).

iv) Voting: Voting is one of the non-trainable combiners used in ensembles of classifiers. Voting operates on labels only, where $d_{t,j}$ is 1 or 0 depending on whether classifier t chooses j, or not, respectively (Tax, Van Breukelen, Duin, & Kittler, 2000; Xu, Krzyzak, & Suen, 1992). The ensemble then chooses class J that receives the largest total vote based on:

a) Majority (plurality) voting - $\sum_{t=1}^{T} d_{t,J}(x) = \max_{j=1,\cdots,c} \sum_{t=1}^{T} d_{t,J}$

b) Weighted majority voting - $\sum_{t=1}^{T} w_t d_{t,J}(x) = \max_{j=1,\cdots,c} \sum_{t=1}^{T} d_{t,J}$

2.5 Summary

The theoretical background that puts the methods and techniques used in the studies reported in this thesis in context is presented in this chapter. The chapter presented a brief introduction to supervised and unsupervised ML approaches. This was followed by an introduction to TM and a description of steps of a typical TM process. A discussion on CV as a means to reduce error in classification performance results and ensure reliability in future prediction was presented with a discussion on ensemble learning as a means to improve the classifiers' performances through a combination of multiple classifiers' results.

The concepts and techniques presented in this chapter contributed to the studies reported in different parts of this thesis. Specifically, the SVM was used for classification throughout; the CV approach and the tf-idf, word2vec, and binary feature representations with the χ^2 method were used in Sections 4.2.1, 5.4 and 6.3. These techniques with ensemble voting technique contribute to the techniques implemented in the tool presented in Chapter 7.

CHAPTER 3

Literature Review

A general overview of ML and TM was presented in Chapter 2. In this chapter, the body of literatures related to this research will be reviewed where the field's usage of most of the subject presented in Chapter 2 will come up. The review takes the form of a mapping study to analyse methods that have been used, how they have been used and the information provided on their usage. The findings of the study presented in this chapter provided the basis for the follow-up studies in the research.

The motivation of this thesis was presented in Section 1.1.5. Particularly, the need for transparency in studies on automatic CS in SRs using TM related techniques is discussed. A mapping study of the literature is presented in this chapter. The aim of the study is to analyse the methods being used and how much information and justification is provided on (the choice of) each of the methods. This covers finding out if:

i) the parameters for the techniques are set in an informed way;

ii) the methods are applied in a statistically valid way - considering data size and method complexity;

iii) the methods are applied in a transparent way to enable independent reproducibility.

Two key discoveries from this study are: a dearth of essential information regarding the choice and use of the TM methods; and an absence of reproduction or replication among the studies. Consequently, the need to investigate the provided information as a source to determine the quality of the models and its usefulness on the reproducibility of the studies.

3.1 Introduction

As pointed out in section 1.1.2, EBSE is anchored on SRs. SRs are useful research tools that seek to locate, collate, analyse and present evidence relating to a specific topic of interest using a rigorous and unbiased approach (Kitchenham et al., 2015). It begins with the development of a review protocol a priori laid down plan on how the review process will be conducted, candidate studies judged and research questions to be answered by the review outcome. This is followed by other stages of the review process terminating with the reporting of the outcome (see Figure 1.1)

A mapping study provides an overview (often visual) of the research that has been conducted and their results (Petersen et al., 2008), rather than perform an in-depth evaluation of the research to answer certain research questions. Mapping studies involve similar processes from protocol development, literature search, citation screening through to reporting (refer to Section 1.1.3), as SRs but because mapping studies are more concerned about providing a structural classification and thematic analysis of the research types and their results rather than evidence synthesis, the quality assessment stage is excluded. Three key differences that exists between a SR and a mapping study are presented below:

i) The primary studies are not evaluated for quality in a mapping study as the goal is not to establish the state of evidence. Therefore, it does not include the research quality assessment step (Petersen et al., 2008; Kitchenham & Charters, 2007).

ii) The mapping study may include more articles than the SR since the articles are not studied as deep as it would have been in a SR (Petersen et al., 2008; Kitchenham & Charters, 2007).

iii) Thematic analysis is a method of choice in a mapping study while meta analysis is favoured in SR (Petersen et al., 2008; Kitchenham & Charters, 2007).

The research reported in this chapter was the first to investigate the technical information provided regarding the models being proposed in different studies and the demonstration of awareness on the limitations of the algorithms adopted given the different data sizes. The research focused specifically on the models proposed for automatic CS using TM techniques.

This study and eventually other experiments in this project will consider research and datasets covering both the medical and SE research on the automation of CS in SRs. The following are the reasons for this choice:

i) These two fields not only have the SR in common as an element of their evidence based research but are also the most active in the quest for automation using TM techniques.

ii) The medical evidence-based research has more labelled data that can be useful for the exploration and development of predictive models needed in the automation research.

iii) The CS process in both fields is the same, at least as far as the initial abstract-title screening is concerned (see Section 1.1.3). Therefore, a predictive tool should be able to learn from its data irrespective of the source of its input source; hence, it should be able to work across the disciplines.

iv) More work has been undertaken in the medical field on this subject than SE.

A SR study (O'Mara-Eves et al., 2015) was published a few months before the mapping study commenced, therefore, the literature search phase was substituted with the adoption of studies included in the SR. The SR (O'Mara-Eves et al., 2015), focussed only on non-technical aspects of the TM techniques used in its selected studies. The mapping study on the other hand focussed on the availability of information related to the TM methods being used, including the description and explanation of the methods, process of setting the parameters, assessment of the appropriateness of their application given the size and dimensionality of the data used, performance on training, testing and validation data sets, and level of reproduction or replication among the studies. 35 out of the 44 papers from the SR were finally included for the mapping study. In order to ensure that all relevant publications following the mapping study were covered, a supplementary review of related literature published since the mapping study was performed to ensure full coverage of relevant literatures.

The study reported in this chapter have been published in the 20th International Conference on Evaluation and Assessment in Software Engineering (Olorisade et al., 2016).

3.2 Automation of the SR process

Automation of the individual stages or the whole process of the SR has continue to attract the attention of researchers. Within software engineering, Marshall and Brereton (2013), Marshall, Brereton, and Kitchenham (2014) undertook research to investigate tools to support SR. There are research also undertaken to automate specific stages in both healthcare and SE domains. A research on automatic study identification and retrieval was published by Ghafari, Saleh, and Ebrahimi (2012). Studies on automatic CS were reviewed by O'Mara-Eves et al. (2015), Tsafnat et al. (2014) while 26 studies on automatic data extraction were reviewed by Jonnalagadda, Goyal, and Huffman (2015).

3.2.1 Complete SR process automation

SLuRp, is a web based tool designed for the management of all types of data involved in the SR process (Bowes, Hall, & Beecham, 2012). It was developed to 'semi-automatically' search and retrieve studies from limited databases, capture data relating to the review being carried out, inclusion/exclusion criteria, reasons for acceptance/rejection, disagreement reconciliation and storage of full copies of included papers. Another tool with similar functionality is SLR-Tool (Fernández-Sáez, Bocco, & Romero, 2010). The tool uses TM techniques to enhance decision making. SLR-Tool can store papers in pdf, communicate with bibliography management software and can also collect and import data to Excel among other functions (Fernández-Sáez et al., 2010). StArt, is a tool reported in the literature for managing all phases of the SR except literature search however, it can read citations in 'BibTex' format. It can rank papers and record information and decisions regarding each paper at different phases of the review process (Hernandes, Zamboni, Fabbri, & Thommazo, 2012). Another addition to these tools is SESRA, a web based SR management tool (Molléri & Benitti, 2015).

Based on information provided in the papers, the major tasks of the SR supported - limited or fully - by each of the tools are presented in Table 3.1. The '•' sign indicates supported feature, indicates otherwise. A more detailed comparative analysis of the features offered by these tools can be found in (Marshall et al., 2014).

Table 3.1: Systematic review phase managed by the tools

SR Stage	SLuRp	StArt	SLR–Tool	SESRA
Protocol development		•		•
Study identification	•	•		•
Study selection	•	•	•	•
Study evaluation	•	•	•	•
Data extraction	•	•	•	•
Data synthesis	•	•	•	•
Reporting	•	•		•

3.2.2 Specific stages automation

A number of studies in recent reviews on methods for SR automation have indicated that there are more studies published on the automation of specific stages of the SR, most especially, CS and data extraction, than on the entire process (Jonnalagadda et al., 2015; O'Mara-Eves et al., 2015). Work in this area is now focused beyond basic - software support development of the SR processes and instead aims to create intelligent system (using Artificial Intelligence methods) that can make independent decisions and therefore reduce the human effort required in SR (Jonnalagadda et al.,

2015; O'Mara-Eves et al., 2015). Study identification, CS and data extraction are the three stages that are currently being focused upon based on available publications. Works undertaken on these three stages are discussed below:

i) Study identification: A federated search tool has been developed to automate searching and retrieval of literature across multiple databases (Ghafari et al., 2012). Tool developers reported promising result from its use across more than 10 databases. However, this tool is not publicly available nor has it been independently evaluated.

ii) Citation screening: This stage has attracted the most attention in terms of an individual SR stage automation (Marshall & Brereton, 2013). The majority of the efforts to automate the CS stage are centred on TM techniques; these are explored in the context of easing the task of selecting the relevant studies from the results of the study search. Forty-four of these studies were reviewed and reported in (O'Mara-Eves et al., 2015). The studies focused on a range of interests, from reducing screening workload to prioritisation of documents for screening. There is no overarching or widely accepted tool/method yet, but results are promising.

iii) Data extraction: A recent review by Jonnalagadda et al. identified 26 studies focused on automation of the data extraction stage in SR (Jonnalagadda et al., 2015). The majority of the studies reviewed also used machine learning techniques for automation.

More recently, Khabsa et al. have undertaken a work on the use of the random forest technique for automatic CS, their method was embedded in a tool for SR named Rayyan (Khabsa et al., 2016). Work was also undertaken by Mo, Kontonatsios, and Ananiadou (2015) on the use of Latent Dirichlet Allocation (LDA)–based document representations to support automatic CS, the use of a similar approach with active learning was reported in (Hashimoto, Kontonatsios, Miwa, & Ananiadou, 2016) and research using the active learning approach were undertaken by Z. Yu et al. (2016), Timsina, Liu, and El-Gayar (2015). Work proposing the use of a semi-supervised approach using label propagation was published in (Kontonatsios et al., 2017). SWIFT-Review is the outcome of work undertaken by Howard et al. (2016). Work using support vector machine and Unified Medical Language System features was undertaken by Timsina, Liu, and El-Gayar (2016).

3.3 The mapping study

As mentioned in Section 3.1, largely mapping studies and SRs share common process, this study was conducted in line with the guidelines proposed by Kitchenham and

Charters (2007). A protocol was initially developed to guide the study process. The protocol was reviewed by the project's supervisory team which had on it a member of the software engineering and systems research group[1] (Prof. Pearl Brereton (OPB)), a member of the Machine Learning (ML) and computational intelligence research group [2] (Prof. Peter Andras (PA)) and another member of the software engineering and systems research group (Dr Ed de Quincey (EDQ)) who was invited only for the purpose of conducting the mapping study. The research questions are presented in this section alongside the description of the process followed to conduct the study.

3.3.1 Research questions

The objectives of the study were to investigate transparency and appropriateness of models proposed for automatically screening of citations during SRs. Transparency refers to the level of information provided which may be useful to support reproducibility of the studies; while appropriateness refers to the indication of awareness on the suitability of the proposed models within the context of the data they were generated from or their constraints or limitations within the same context. The study was focused on automatic CS models using TM techniques, models from other techniques or studies on the automation of other stages of the SR process were not considered. Three research questions were created to address the objectives of this study:

1. RQA: What information is available on the use and distribution of specific TM algorithms being proposed to automate CS in SR - How well are the algorithms used described and/or justified in the context of use, what information is provided about the data size and to what extent is the effect of data size on the TM algorithms used taken into account?
2. RQB: What is the proportion of the included (positive example)/excluded (negative example) documents and how did the classifiers perform during training, validation and testing given the metrics used?
3. RQC: How comparable are the results of the different studies reviewed?

3.3.2 Search strategy

As previously pointed out in the introduction to this chapter, this study did not involve any new search for articles rather the included articles from the process of an existing SR of TM based CS studies (O'Mara-Eves et al., 2015) were adopted. The adoption of articles from the SR was considered appropriate because it was recent at the time of conducting the mapping study and it was also comprehensive. The O'Mara-Eves

[1] https://www.keele.ac.uk/scm/research/compsci/softwareandsystemsengineering/

[2] https://www.keele.ac.uk/scm/research/compsci/machinelearningandcomputationalintelligence/

et al. (2015) review selected papers on TM methods or metrics that were applied to the screening stage of a SR (or similar evidence review), however, the study did not look at the methods in any depth since their intended audience were users of the technologies rather than computer scientists.

3.3.3 Study selection criteria

Irrespective of the fact that the study adopted articles from an existing review, further criteria were set to ensure only articles relevant to the mapping study were included and others excluded:

- Inclusion criteria:
  - The publication must be reporting the outcome of a research exercise/experiment/case study/development.
  - The topic of discussion or field of application must relate to ML algorithm based classification model using TM technique.
  - The context of use must be CS in SR.
- Exclusion criteria:
  - Communications/opinion papers.
  - Natural language technique studies not using ML algorithm.
  - Information retrieval or information extraction studies.

In order to avoid duplication, studies reported across multiple publications are considered together and where papers report multiple studies, the studies are considered separately. The screening was done by myself since this process is considered a secondary screening in the mapping study.

3.3.4 Data extraction

The following data were extracted from each paper:

a) General
   i) Study type
   ii) Bibliographic information
   iii) Study objective

b) Data information
   i) Data source
   ii) Corpus size

iii) Feature set composition

c) Feature representation and dimensionality reduction

i) Feature vector
ii) Preprocessing information
iii) Feature selection technique
iv) Final feature size

d) Model choice

i) Classifier
ii) Variant algorithm
iii) Third party tool
iv) Proposed method (by author)

e) Training and assessment

i) Modelling approach
ii) Positive/negative sample ratio in training data/test
iii) Performance measure (train/test)

The review team consisted of four reviewers. Myself (BKO) as the lead reviewer, extracted data from all the articles. The articles were randomly divided amongst the other three reviewers (see Section 3.3) for data extraction except the bibliographic information. The extracted data was stored using the Microsoft Excel application, while the bibliographic information were stored with Mendeley [3]. A pilot study was initially conducted to assess the Excel form and reviewers' understanding of its fields. The extraction form was modified after the exercise to correct the inconsistencies identified. After the full data extraction, differences in the extracted data were resolved through two meetings involving all the reviewers. Any outstanding differences were resolved through meetings between the lead reviewer and the other review team member concerned. No situation warranted inviting a third reviewer to mediate in any of the latter resolution meetings.

3.4 Results

3.4.1 Data extraction

Each of the papers was identified by a Paper ID, and a Study ID to differentiate where a paper reported the results of more than one study. Eight of the 44 articles

[3] https://www.mendeley.com/reference-management/reference-manager

were excluded following study selection process because they did not fully meet one or more of the inclusion criteria for the study while one was unpublished bringing the total to nine. Three of the papers (P07, P28 and P36) were excluded because they are communication between different research teams as a follow up discussion on their previous studies' results. Though, SR was discussed in P33, the technique used was not ML based; no TM experiment was conducted in P44. Two of the studies (P16 and P17) were excluded because the techniques used were neither ML based nor applied within the SR context. Another paper (P31) was excluded because the focus of the study was on the performance of different feature selection techniques and not the classification model. An unpublished article (P35) included in the original review could not be retrieved. The list of excluded papers is presented in Appendix A.1.

The total number of papers included was 35 with a total of 45 studies. Multiple studies were recorded in serial numbers 1, 2, 10 and 35 (Table 3.2).

Table 3.2: List of included papers

Paper ID	**Study ID**	**Paper Title**	**Paper Reference**
P01	S01 S02 S03	Towards automating the initial screening phase of a systematic review	Bekhuis and Demner-Fushman (2010)
P02	S04 S05 S06 S07 S08	Screening non-randomized studies for medical systematic reviews: A comparative study of classifiers	Bekhuis and Demner-Fushman (2012)
P03	S09	Feature engineering and a proposed decision-support system for systematic reviewers of medical evidence	Bekhuis, Tseytlin, Mitchell, and Demner-Fushman (2014)
P04	S10	Combining relevancy and methodological quality into a single ranking for evidence-based medicine	S. Choi, Ryu, Yoo, and Choi (2012)
P05	S11	Reducing workload in systematic review preparation using automated citation classification	A. M. Cohen et al. (2006)
P06	S12	An effective general purpose approach for automated biomedical document classification	A. M. Cohen (2006)
P08	S13	Optimizing feature representations for automated systematic review work prioritization	A. M. Cohen (2008)

Table 3.2: List of included papers (continued)

Paper ID	Study ID	Paper Title	Paper Reference
P09	S14	Cross-Topic Learning for Work Prioritization in Systematic Review Creation and Update	A. M. Cohen, Ambert, and McDonagh (2009)
P10	S15	Studying the potential impact of automated document classification on scheduling a systematic review update	A. M. Cohen, Ambert, and McDonagh (2012)
P11	S16 S17	A Pilot Study Using Machine Learning and Domain Knowledge to Facilitate Comparative Effectiveness Review Updating	Dalal et al. (2013)
P12	S18	A Prospective Evaluation of an Automated Classification System to Support Evidence-based Medicine and Systematic Review	A. M. Cohen, Ambert, and McDonagh (2010)
P13	S19	A visual analysis approach to validate the selection review of primary studies in systematic reviews	Katia R Felizardo, Andery, Paulovich, Minghim, and Maldonado (2012)
P14	S20	Using Visual Text Mining to Support the Study Selection Activity in Systematic Literature Reviews	Katia R Felizardo et al. (2011)
P15	S21	The use of visual TM to support the study selection activity in systematic literature reviews: A replication study	Katia Romero Felizardo, Souza, and Maldonado (2013)
P18	S22 S23	Building systematic reviews using automatic text classification techniques	Frunza, Inkpen, and Matwin (2010)
P19	S24 S25	Exploiting the systematic review protocol for classification of medical abstracts	Frunza, Inkpen, Matwin, Klement, and Oblenis (2011)
P20	S26	Automatic text classification to support systematic reviews in medicine	Adeva, Atxa, Carrillo, and Zengotitabengoa (2014)
P21	S27	A New Iterative Method to Reduce Workload in the Systematic Review Process	Jonnalagadda and Petitti (2013)

Table 3.2: List of included papers (continued)

Paper ID	Study ID	Paper Title	Paper Reference
P22	S28	Improving the performance of text categorization models used for the selection of high quality articles	Kim and Choi (2012)
P23	S29	Using classifier performance visualization to improve collective ranking techniques for biomedical abstracts classification	Kouznetsov and Japkowicz (2010)
P24	S30	Classifying biomedical abstracts using committees of classifiers and collective ranking techniques	Kouznetsov et al. (2009)
P25	S31	Text classification on imbalanced data: Application to Systematic Reviews Automation	Yimin Ma (2007)
P26	S32	A Visual Text Mining approach for Systematic Reviews	Malheiros, Hohn, Pinho, and Mendonca (2007)
P27	S33	Facilitating biomedical systematic reviews using ranked text retrieval and classification	Martinez, Karimi, Cavedon, and Baldwin (2008)
P29	S34	A new algorithm for reducing the workload of experts in performing systematic reviews	Matwin, Kouznetsov, Inkpen, Frunza, and O'blenis (2010)
P30	S35	Reducing systematic review workload through certainty-based screening	Miwa, Thomas, OMara-Eves, and Ananiadou (2014)
P32	S36	Pinpointing needles in giant haystacks: use of TM to reduce impractical screening workload in extremely large scoping reviews	Shemilt et al. (2014)
P34	S37	Linked data approach for selection process automation in systematic reviews	Tomassetti et al. (2011)
P37	S38	Who should label what? Instance allocation in multiple expert active learning	Wallace, Small, Brodley, and Trikalinos (2011)

Table 3.2: List of included papers (continued)

Paper ID	Study ID	Paper Title	Paper Reference
P38	S39	Toward modernizing the systematic review pipeline in genetics: efficient updating via data mining	Wallace, Small, Brodley, Lau, Schmid, et al. (2012)
P39	S40	Deploying an interactive machine learning system in an evidence-based practice center	Wallace, Small, Brodley, Lau, and Trikalinos (2012)
P40	S41	Modelling Annotation Time to Reduce Workload in Comparative Effectiveness Reviews Categories and Subject Descriptors Active Learning to Mitigate Workload	Wallace, Small, Brodley, Lau, and Trikalinos (2010)
P41	S42	Active Learning for Biomedical Citation Screening	Wallace, Small, Brodley, and Trikalinos (2010)
P42	S43	Semi-automated screening of biomedical citations for systematic reviews	Wallace, Trikalinos, Lau, Brodley, and Schmid (2010)
P43	S44 S45	GAPscreener: an automatic tool for screening human genetic association literature in PubMed using the support vector machine technique	W. Yu et al. (2008)
P07, P16, P17, P28, P31, P33, P35, P36 and P44 were excluded (see Appendix A.1)			

3.4.2 Algorithms: usage, information and justification

This section addresses RQA: *What information is available on the use and distribution of specific TM algorithms being proposed to automate CS in SR – How well are the algorithms used described and/or justified in the context of use, what information is provided about the data size and to what extent is the effect of data size on the TM algorithm used taken into account?*

Support vector machine (SVM) was the most used algorithm. It was used in 31% of the studies, excluding its usage in Ensemble of classifiers, and has been used in at least one experiment annually since 2006 (see Table 3.3). Ensemble of classifiers was used in 22% (see Table 3.3 with pictorial representation in Figure 3.1) while Naïve Bayes (NB) was used in 14% of the studies. About 50% of the studies tried and reported more than one classifier. Their usage in the papers reviewed including

other algorithms used is presented in Table 3.4.

Table 3.3: Classification algorithm used by year

Algorithm	2006	2007	2008	2009	2010	2011	2012	2013	2014	Total	%
SVM	1	1	3	1	4	1	4		1	16	31%
EvoSVM					1		1			2	4%
NB		1			1	1	2		2	7	14%
cNB					1	1	1			3	6%
KNN						1	1		1	3	6%
k-Means						1	1			2	4%
Decision Tree		1			1					2	4%
WAODE					1					1	2%
NN	1									1	2%
Ensemble	1			2	3	1	2	1	1	11	22%
Regression							1			1	2%
Rocchio									1	1	2%
D. Semantics								1		1	2%

Apart from the individual techniques, different variant options have been tried as shown in Table 3.5. Less than 20% of the studies explained the algorithms they used and provided some justification why the particular algorithm was chosen over others in the context of their studies. None of the studies that used variants of an SVM classification algorithm or optimisation settings, e.g., kernels, C or gamma values, justified or provided insights into why they chose one option over others.

In 70% of the cases, the studies reported using open access ML implementation frameworks like Weka (M. Hall et al., 2009) with different settings, mostly the default, without discussing why they (the settings) were suitable within the context of their own experiment(s).

3.4.2.1 Data size

The summary of the corpus sizes used in the studies is presented in Figure 3.2. None of the papers considered the impact of the corpus size on the statistical appropriateness of the application of the ML methods that they used. In particular, the papers describing the application of SVM did not report the number of SV in the final classifier, which is critical information to confirm that over-fitting by the classifier was

Table 3.4: Classification algorithms used in different papers

S/N	Algorithm	Papers
1	SVM	P04, P06, P08, P09, P10, P12, P20, P22, P25, P27, P37, P39, P40, P41, P42, P43
2	EvoSVM	P01, P02
3	Naïve Bayes (NB)	P02, P03, P04, P20, P25, P29, P34
4	K-Nearest Neighbour	P02, P14, P20
5	K-Means	P13, P14
6	Complement Naïve Bayes (cNB)	P02, P18, P19
7	Decision Tree (DT)	P01, P25
8	WAODE	P01
9	Neural Network (NN)	P05, P11
10	Regression	P11
11	Ensemble	P30, P32, P38, P6, P31, P18, P42, P19, P23, P24
12	Rocchio	P20
13	Distributional semantics with relevance feedback	P21

avoided.

3.4.2.2 Feature representation

Except where explicit information was not provided, all the studies used the vector space model – 'Bag-of-Words (BOW)', for feature representation (Korde & Mahender, 2012; Kumar & Chandrasekhar, 2012). Frequency based representations was the most used while seven have used binary feature representation (Table 3.6). Some studies also experimented with multiple n-grams (Bekhuis & Demner-Fushman, 2012; Bekhuis et al., 2014; A. M. Cohen, 2008; A. M. Cohen et al., 2012, 2010).

3.4.2.3 Feature selection techniques

Feature selection (FS) techniques used across the studies are: term frequency (tf), term frequency-inverse document frequency (tf-idf), information gain (IG), Okapi

Table 3.5: Classifier variants usage

S/N	Classifier	Variant	Studies	Respective years
1	SVM	Linear Kernel	P04, P37, P41	2012, 2011, 2010
		Radial Basis Function kernel	P01, P04, P32, P43	2010, 2012, 2013, 2008
		Polynomial Kernel	P04	2012
		Sigmoid	P04	2012
		Epanechnikov (degree 3, 4)	P01	2010
		Active Learning	P30, P37, P39, P40	2014, 2011, 2012, 2010
2	KNN	K = 1	P02	2012
3	Naïve Bayes	Multinomial	P2, P4	2012, 2012
		Complimentary	P02, P03	2012, 2014
4	Neural networks	Voting Perceptron	P05	2006
		Generalized Linear Model	P11	2012
5	Regression	Gradient boosting machine	P11	2012
6	Ensemble	Voting	P06, P18, P19	2011, 2006, 2010
		Bagging	P32, P38	2013, 2012
		Unspecified	P24, P30, P42	2009, 2009, 2014, 2010
		Query by Committee	P23	2010

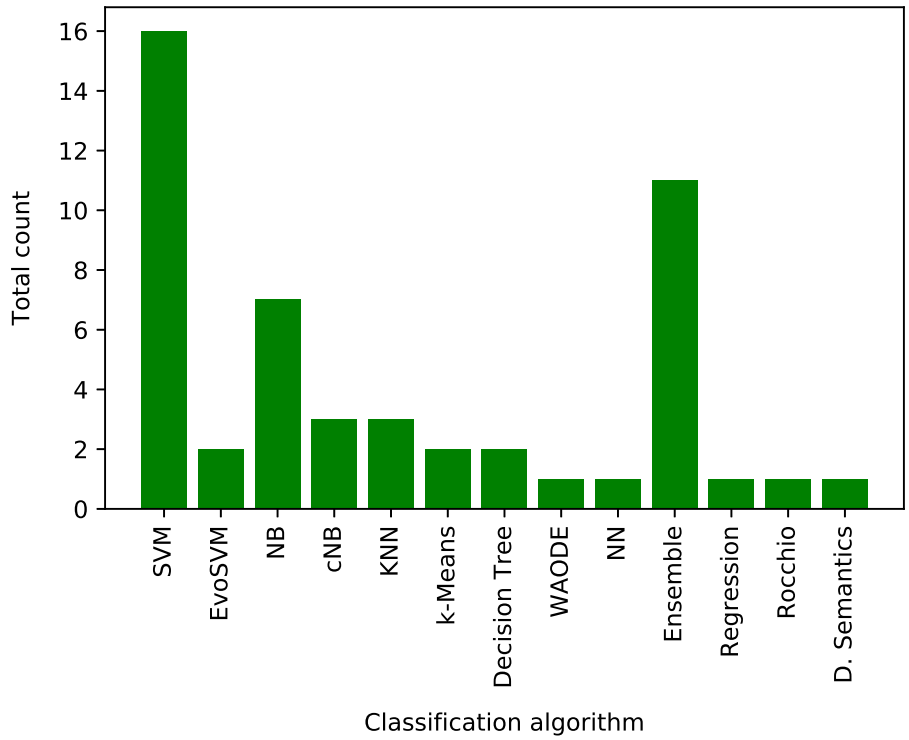

Figure 3.1: Number of classifiers used in the studies

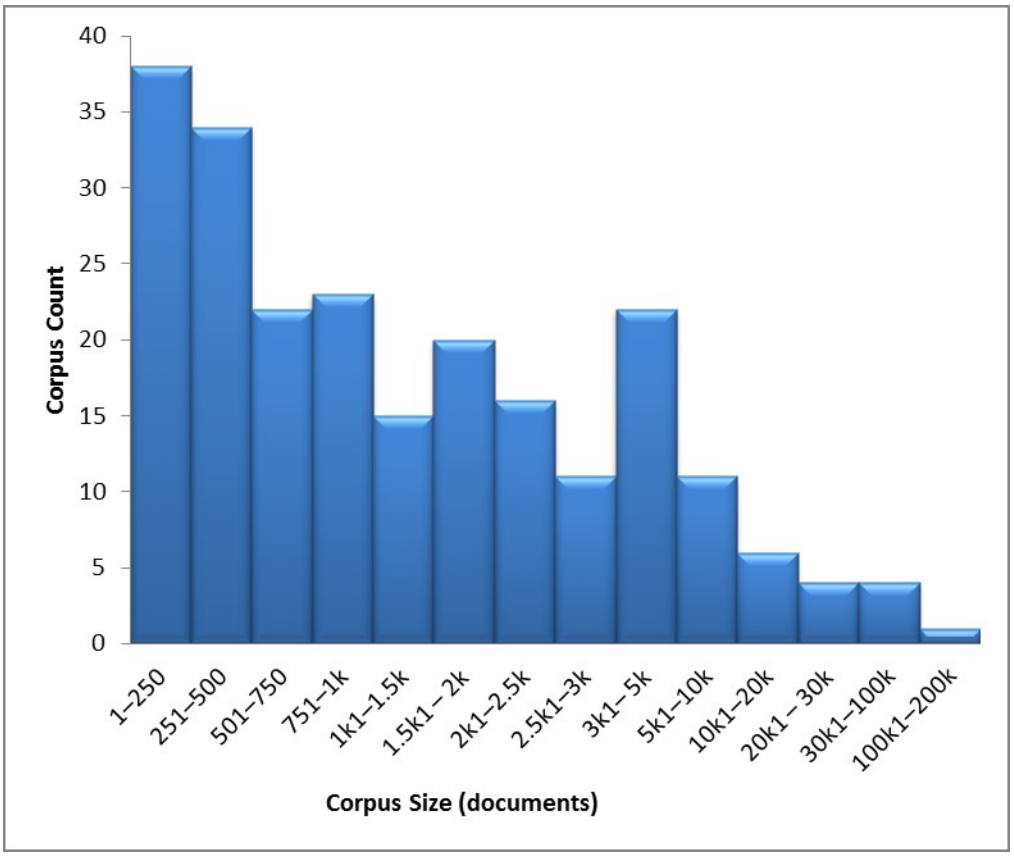

Figure 3.2: Corpus size range used across all studies[a]

[a] In the figure label, yk1 $\implies y * 1000 + 1$.

Table 3.6: Feature representation techniques usage

S/N	Feature representation technique	Count
1	Term frequency based	25
2	Binary vector	7
3	SOSCO	2
4	No Explicit Information	4

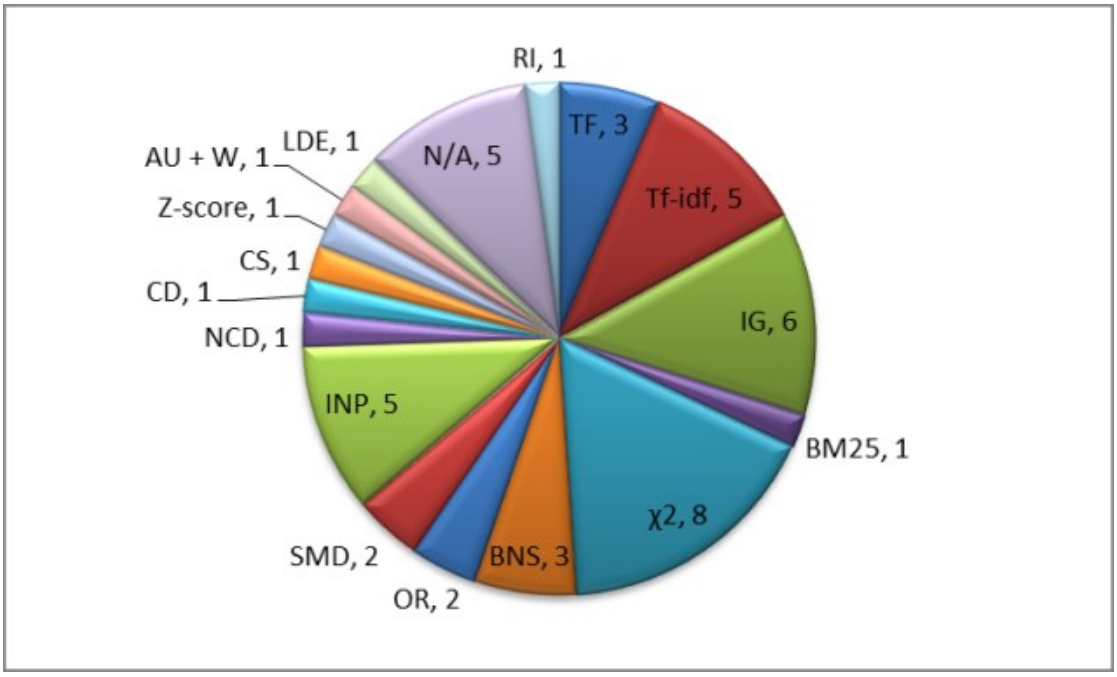

Figure 3.3: Feature selection/extraction techniques distribution

BM25 (BM25), bi-normal separation (BNS), odds ratio (OR), signed margin distance (SMD), normalized compression distance (NCD), cosine distance (CD), covariate shift (CS), aggressive under sampling + weighting (AU + W), linked document enrichment (LDE) and random indexing (RI). The techniques and the number of times each was used across all the studies is presented in Figure 3.3. There are situations where studies did not provide information concerning how FS was handled, 'INP' was used to signify such in Figure 3.3, whilst 'NA' implies 'Not Applicable', for situations with no information. About 50% of the studies used multiple techniques to compare performance. Feature extraction approach was rarely used, LDA was used in (Miwa et al., 2014) and topic modelling in (Bekhuis et al., 2014).

3.4.2.4 Proposed tools and algorithms

Some of the studies have proposed novel tools, approaches or algorithms. An SVM based tool called GAPScreener was proposed by W. Yu et al. (2008), ABSTACKR,

an Active Learning based system was proposed in (Wallace, Small, Brodley, Lau, & Trikalinos, 2010, 2012). A ranking algorithm was proposed in (Kouznetsov et al., 2009; Kouznetsov & Japkowicz, 2010), while an approach tagged 'ranked-retrieval-re-rank' was proposed in (Martinez et al., 2008); a factorized form to cNB was proposed in (Matwin et al., 2010). Tomassetti et al. (2011) proposed an enriched approach for feature selection based on linked data. A 'metacognitive Multiple Experts Active Learning (MEAL)' algorithm was proposed in (Wallace et al., 2011).

3.4.2.5 Third party frameworks

Machine learning toolboxes were used to carry out the experiments reported in 28 of the papers. The main toolboxes used are: WEKA (M. Hall et al., 2009), Projclus (Paulovich & Minghim, 2006), Revis, PEx tool [4], Pimiento (Adeva & Calvo, 2006), RapidMiner [5], LibSVM [6] and SVMLight [7].

3.4.3 Class imbalance and classifier performance

This section addresses RQB: *What is the proportion of the included (positive example)/excluded (negative example) documents and how did the classifiers perform during training, validation and testing given the metrics used?*

The average percentage ratio of the positive to negative examples in the corpus used for 90% or more of the studies is 10%:90%. The studies tried to maintain this ratio in the training and test data (stratified sampling). This issue of class imbalance was handled in different ways across the studies, a summary of the different approaches used was reported in (O'Mara-Eves et al., 2015).

The majority of the studies used a Cross Validation (CV) approach for building the models. This was as a result of the relatively small sizes of the datasets (see Figure 3.2) used across the studies. The 5×2-fold CV was used in (Bekhuis et al., 2014; A. M. Cohen et al., 2010; A. M. Cohen, 2008; A. M. Cohen et al., 2012; A. M. Cohen et al., 2006; Kouznetsov et al., 2009; Matwin et al., 2010), 10-fold CV was used in (Bekhuis & Demner-Fushman, 2012, 2010; S. Choi et al., 2012; Adeva et al., 2014; Kouznetsov & Japkowicz, 2010; Tomassetti et al., 2011; Wallace et al., 2011) and 5-fold CV was used in (Dalal et al., 2013). Kouznetsov et al. (2009), Kouznetsov and Japkowicz (2010) used both 5×2 and 10-fold CV with stratified random sampling; multiple n-way CV with n ranging between 2 to 256 increasing by power of 2 was used in (A. M. Cohen et al., 2009) and cost rejection sampling was used in (A. M. Cohen, 2006).

[4] http://infoserver.lcad.icmc.usp.br/infovis2/PEx
[5] https://rapidminer.com/
[6] https://www.csie.ntu.edu.tw/ cjlin/libsvm/
[7] http://svmlight.joachims.org/

In terms of performance metrics, (mean) recall, (mean) precision, (mean) F and the area under the receiver operating characteristics curve (AUC) were mostly used. High recall implies few false negatives in the result while high precision implies few false positives. The F-measure is a weighted harmonic mean assessing the precision-recall trade-off and AUC is the probability that a model will rank a randomly chosen positive sample higher than a randomly chosen negative sample. Mean recall was 95% and above in (Bekhuis & Demner-Fushman, 2012, 2010; A. M. Cohen et al., 2006; Frunza et al., 2011; Kouznetsov & Japkowicz, 2010; Matwin et al., 2010; Tomassetti et al., 2011; W. Yu et al., 2008; Wallace, Small, Brodley, Lau, Schmid, et al., 2012) while it was below 95% in (S. Choi et al., 2012; Kouznetsov et al., 2009). Precision on the other hand was over 10% in (Bekhuis & Demner-Fushman, 2012, 2010; W. Yu et al., 2008; A. M. Cohen et al., 2006; Frunza et al., 2011; Kouznetsov et al., 2009; Kouznetsov & Japkowicz, 2010). AUC was used in (A. M. Cohen, 2008; Dalal et al., 2013; Kouznetsov & Japkowicz, 2010; Martinez et al., 2008; Miwa et al., 2014; W. Yu et al., 2008) and the result was over 0.5 in all the studies. A. M. Cohen et al. (2006) proposed the WSS metric based on the amount of manual work saved. This measure was also used in (Frunza et al., 2010; Jonnalagadda & Petitti, 2013; Martinez et al., 2008; Matwin et al., 2010) to determine how much manual screening effort was saved given the classification result. Training performance was mostly sustained during testing or CV.

3.4.4 Result comparability

This section addresses RQC: *How comparable are the results of the different studies reviewed?*

The datasets used in more than one paper are presented in Table 3.7 along with the classifiers and metrics used. Where classifiers are compared in a study '$>$' is used to denote 'better than' and '$\approx$' used to denote 'equal or similar' in respect of reported performance values, otherwise, the classifier used is just listed under comment. The table is not presented for the purposes of comparison but to gain insight into study variability based on dataset, metrics and classification model. It can be inferred from the extent of variability in metrics and techniques (comment) in Table 3.7 that datasets are being reused without any actual relation to the results (and/or process) of previous experiments that had used the same data.

Comparing the performance of classifiers from different experimental settings is not trivial in ML. The performance of classifiers is usually specific to the context of use, thus, it is not easy to compare classifiers trained and used on different datasets (Sebastiani, 2002) or from different experiments (Baharudin, Lee, & Khan, 2010; S. B. Kotsiantis et al., 2006). It may be possible to compare, when the same dataset is used for different classifiers in different experiments, but if, for example, the dataset

was not split in exactly the same way the comparison is still questionable. This is the case with most of the studies reviewed in this study; a few of them used the same dataset in their experiments (Table 3.7), and there was no record of whether a replica of the training set and test set in an experiment was repeated in another. Based on the used of same dataset, some researchers have attempted to compare results with other studies used (A. M. Cohen et al., 2006; A. M. Cohen, 2011; Jonnalagadda & Petitti, 2013; Matwin, Kouznetsov, Inkpen, Frunza, & O'blenis, 2011). None of the studies provided explicit information on which portion of their dataset was used as training and test portions or in each fold of CV as the case may be.

Table 3.7: Studies with common dataset

S/N	Dataset	Paper ID	Metrics	Comment
1	DERP	P4	AUC	SVM
		P29	WSS@95%	FCNB
		P22	accuracy	SVM
		P08	AUC	SVM
		P01	Recall, precision, F1	EvoSVM > WAODE > NB
		P10	Recall, precision, F	SVM
		P27	WSS, AUC	SVR
		P06	Un	SVM
		P21	WSS	Relevance feedback
		P5	Recall, precision, F1, WSS	Perceptron
		P08, P09, P12	AUC	SVM
2	TrialStat SR	P18	Recall, precision, F, WSS	cNB
		P19	Recall, precision, F	SVM ≈ NB
		P31		Ensemble
		P24	Recall, precision, workload save	Ensemble
		P23	False negatives	Ensemble
3	COPD	P37	U19	MEAL (SVM) > PAL (SVM)
		P41	U19	SVM(coFeature) > (Simple) > (Random) > (Features Simple)
		P42	Yield, burden	SVM (AL)

Table 3.7: Studies with common dataset (continued)

S/N	Dataset	Paper ID	Metrics	Comment
		P30	Utility, coverage, AUC	Ensemble SVM
4	Proton beam	P40		SVM (AL)
		P41	U19	SVM(coFeature) > (Simple) > (Random) > (Features Simple)
		P42	Yield, burden	SVM Ensembles
5	Micro nutrients	P30	Utility, coverage	Ensemble SVM
		P41	U19	SVM(coFeature) ≈ (lp) > (Random) > (Simple) > (Features Simple)
		P42	Yield, burden	SVM (AL)
		P30	Utility, coverage	Ensemble SVM

3.4.5 Threats to study validity

Some of the design decisions impose limitations on the construct validity of the study from the possibility of incomplete identification of existing research because no independent literature search was conducted. As mentioned in the introduction of this chapter, the articles evaluated in this study are limited to those included in a previous SR. The study results therefore, are affected by the completeness of the published SR. However, the articles reviewed in the update search did not cite or report any article published prior to the conclusion of the review and not reviewed in it. However, since the articles adopted from the SR were not updated through a new search to include any other relevant articles published since February 2014 (a gap of one year) for the mapping study, it is likely, that some articles are missed. However, relying on a SR which has a more stringent searching and study identification requirement than a mapping study (Kitchenham et al., 2010), it is possible that the included articles are representative of the field. A supplementary literature update has since been conducted to ensure all relevant articles has been considered between the period the mapping study was completed and the submission of this thesis (see Section 3.5).

Another threat imposes an internal validity on this study from the possibility of bias during the process of agreeing on values of extracted data. As mentioned in Section 3.3.4, there arose situations where there were disagreements in the extracted data between BKO and each of OPB, PA and EDQ. Although, the disagreements

were resolved locally between BKO and each team member, the fact that one of the reviewers is a PhD student with two professors and a lecturer might have influenced the outcome. Every effort was however made to avoid this by ensuring different parties concerned were given the chance to make an explicit presentation of their understanding from which a common agreement was sought.

3.5 Literature update

Other research has been undertaken in this field by other researchers after the mapping study was concluded. Twelve research articles consisting of three secondary and nine primary studies have been further identified and are reviewed in this section (see Table 3.8).

Three of the articles are some form of review or analysis of existing techniques or metrics. A research white paper exploring the potentials of TM techniques in the automation of the SR or any of its sub-processes was published by the United States' Agency for Healthcare Research and Quality (AHRQ) (Paynter et al., 2016). A SR of current methods and metrics being used in automatic CS was undertaken by (Saha, Ouzzani, Hammady, & Elmagarmid, 2016) and a comparative analysis of semi-supervised approaches being explored in CS studies was undertaken by J. Liu, Timsina, and El-Gayar (2016). These are excluded from further analysis as they would not have met the inclusion criteria set for the mapping study.

The findings from these studies is still consistent with the initial mapping study. The SVM algorithm continue to dominate, it was used in five out of the nine primary studies (Timsina et al., 2015; Z. Yu et al., 2016; Hashimoto et al., 2016; Timsina et al., 2016; Mo et al., 2015). Document prioritisation was used in (Howard et al., 2016) whilst the random forest (ensemble) algorithm was explored in (Khabsa et al., 2016). Semi-supervised learning approach to learning seem to be gaining more popularity either through label propagation and/or active learning methods (Kontonatsios et al., 2017; Timsina et al., 2015; Z. Yu et al., 2016; Timsina, Liu, El-Gayar, & Shang, 2016; Hashimoto et al., 2016).

The BOW model using the tf-idf weighting method was used for feature representation in five studies (Z. Yu et al., 2016; Howard et al., 2016; Timsina et al., 2015, 2016; Timsina et al., 2016). The LDA was used to select and represent the features in (Mo et al., 2015; Howard et al., 2016) and topic modelling with a neural network vector space representation was used in (Hashimoto et al., 2016). The spectral embedding technique was used by Kontonatsios et al. (2017) for feature representation.

In terms of models assessment techniques, studies appeared to be shifting focus to measures that are indicative of the amount of time and effort saved rather than recall and precision. The work saved over sampling was used in four studies (Khabsa et al., 2016; Howard et al., 2016; Timsina et al., 2015; Timsina et al., 2016). Yield, burden

Table 3.8: List of updated papers

S/N	Paper Title	Paper Type	Paper Reference
1	Supporting systematic reviews using LDA–based document representation	Primary study	Mo, Kontonatsios, and Ananiadou (2015)
2	SWIFT-Review: a text-mining workbench for systematic review	primary study	Howard et al. (2016)
3	How to Read Less: Better Machine Assisted Reading Methods for Systematic Literature Reviews	Primary study	Z. Yu, Kraft, and Menzies (2016)
4	Active Learning for the Automation of Medical Systematic Review Creation	Primary study	Timsina, Liu, and El-Gayar (2015)
5	Topic detection using paragraph vectors to support active learning in systematic reviews	Primary study	Hashimoto, Kontonatsios, Miwa, and Ananiadou (2016)
6	A semi–supervised approach using label propagation to support citation screening	Primary study	Kontonatsios et al. (2017)
7	Using Semi–supervised Learning for the Creation of Medical Systematic Review: An exploratory Analysis	Primary study	Timsina, Liu, El-Gayar, and Shang (2016)
8	Advanced analytics for the automation of medical systematic reviews	Primary study	Timsina, Liu, and El-Gayar (2016)
9	Learning to identify relevant studies for systematic reviews using random forest and external information	Primary study	Khabsa, Elmagarmid, Ilyas, Hammady, and Ouzzani (2016)
10	EPC methods: an exploration of the use of text–mining software in systematic reviews	Secondary study	Paynter et al. (2016)
11	A large scale study of SVM based methods for abstract screening in systematic reviews	Secondary study	Saha, Ouzzani, Hammady, and Elmagarmid (2016)
12	A comparative analysis of semi–supervised learning: The case of article selection for medical systematic reviews	Secondary study	J. Liu, Timsina, and El-Gayar (2016)

and Utility are three metrics proposed by Wallace, Small, Brodley, Lau, and Trikalinos (2010). The yield and burden Wallace, Small, Brodley, and Trikalinos (2010) were used in (Hashimoto et al., 2016; Kontonatsios et al., 2017), the two metrics were used in combination with Utility in (Kontonatsios et al., 2017). Recall and precision

were however used in (Mo et al., 2015; Timsina et al., 2015). The precision-recall curve, receiver operating curve, F1 and accuracy were used in (Mo et al., 2015). The recall vs studies reviewed curve was used in (Z. Yu et al., 2016).

Apache Lucene and Termine are the two third party tools used in (Mo et al., 2015).

The datasets used in more than one study (considering them together with the mapping study in this case) are: COPD used in (Hashimoto et al., 2016; Mo et al., 2015; Kontonatsios et al., 2017); the Evidence-based Practice Center (EPC) review data used in (A. M. Cohen et al., 2006) was either fully or partly used in (Z. Yu et al., 2016; Timsina et al., 2015; Timsina et al., 2016; Timsina et al., 2016) and the Protonbeam used in (Mo et al., 2015).

The work of Khabsa et al. was incorporated as part of Rayyan - a tool for conducting SRs (Khabsa et al., 2016; Ouzzani et al., 2016), Zhe Yu et al. packaged their approach into a CS tool called Fastread (Z. Yu et al., 2016) and Howard et al. package their document prioritisation approach into a tool they called SWIFT_Review, which they proposed as a TM framework for SRs (Howard et al., 2016).

Common datasets (4 from the 15 EPC review dataset used by Cohen et al.) and metrics used provided Timsina et al. (2016) the ground to compare their result (recall and precision) with that of A. M. Cohen et al. (2006). Z. Yu et al. (2016) extended the datasets for their experiment beyond software engineering datasets to include the 15 EPC review datasets used by A. M. Cohen et al. (2006) to enable them compare the performance of their model with Cohen et al.'s. Hashimoto et al. (2016) has also compared their results to the work of Miwa et al. (2014).

3.6 Replication/reproduction practice

Replication or reproduction issues were not actually part of the mapping study's primary objectives, it became hard to ignore them given the number of studies reviewed and the absence of independently replicated or reproduced studies or research teams building on the work of others.

Replication/reproduction of experiments is an established practice in science and engineering to underpin theories and techniques, especially in a growing field of research (Basili, Shull, & Lanubile, 1999; Olorisade, Vegas, & Juristo, 2013). This principle has also been recognized and encouraged in software engineering demonstrated by the existence of research groups with 'empirical' or 'evidence based' attached to their names (González-Barahona & Robles, 2012). Study reproduction with the same, similar or different datasets are useful to verify, extend or complement existing results (Vegas, Juristo, Moreno, Solari, & Letelier, 2006; F. J. Shull, Carver, Vegas, & Juristo, 2008).

Considering the nascent stage of SR in SE and the application of TM to the auto-

mation of some of its stages, it is thus important for independent research teams to reproduce published studies in whole or part as a means to establish efficiency, maturity and applicability of proposed methods and techniques (J. Miller, 2005; F. J. Shull et al., 2008).

Replications in the studies reviewed were often conducted by the same research groups. One such in-team replication undertaken by Wallace et al. led to the creation of a tool tagged ABSTRACKR (Wallace, Small, Brodley, Lau, & Trikalinos, 2010, 2012). An independent evaluation of ABSTRACKR was undertaken in (Rathbone et al., 2015).

It was noted that some of the studies have been attempting some form of replication; in fact six datasets were found to be used by more than one study (see Table 3.7). In addition, Cohen et al. and Matwin et al.'s teams are already comparing model results based on use of the same dataset (A. M. Cohen, 2008, 2011; Matwin et al., 2010, 2011), Felizardo et al. have also replicated their study (Katia R Felizardo et al., 2011; Katia Romero Felizardo et al., 2013). However, more research needs to be done given the fact that SR is now increasingly being adopted across disciplines from medicine to social science, SE and computer science. In order to build useful tools, research teams may require access to data used in studies from other disciplines which may not be as readily available compared to data from within the discipline. More comparisons to existing results were found in Timsina et al. (2016), Z. Yu et al. (2016) which compared the result of their research to that of A. M. Cohen et al. (2006), Matwin et al. (2010). However, the variables that change from one study to the other were often too many for a consistent replication. What is currently being witnessed is more of a result comparison than a progressive replication of studies.

3.7 Discussion

The results of this study serve as a basis for the programme of research reported in this thesis. In this section, the implications of the literature review are discussed. Summarily, the goals of the mapping study were:

i) To analyse the TM methods being used in CS automation studies

ii) To investigate how much information were reported about these methods

iii) To establish any justification provided for the choice of the methods

The SVM algorithm has been reported to have the advantage of coping with high dimensional data without significant impact from class imbalance (S. B. Kotsiantis et al., 2006). It is less affected by the size of its input and requires moderate samples for training (Baharudin et al., 2010; S. B. Kotsiantis et al., 2007; Sebastiani, 2002).

It is also suited to high feature to low training instance situation (Ikonomakis et al., 2005). These facts might have accounted for the performance recorded and substantial use of the support vector machine in the studies. Attempts to ensure more reliable classification performance results might have accounted for the high use of ensemble methods as well. Naïve Bayes on the other hand did not perform well.

The corpus sizes used across the studies as shown in Figure 3.2 suggest that the majority of the experiments used corpus sizes that calls into question the statistical reliability of the classification model built through such corpus. There was rarely any justification across all the studies for the different decisions about the choice of a certain technique or approach within the context of use.

Insufficient information made it hard to assess the process and statistical validity of the majority of the studies, for example, none of the studies that used SVM reported the number of SVs they found. Similarly, in the case of the application of neural networks, there was no information on the number of neurons or hidden layers used. Thus, it is hard to judge how overfitting was controlled and to what extent the complexity of the classifier was considered. There was no mention of the bias/variance trade-off characteristics of the classification algorithms and the impact of the data size in this context. The role data size plays in learning, generalisation ability and classification performance of a model was not emphasized in any of the studies. Notably, the positive to negative example ratio with the number of effective parameters (complexity) is quite important to determine the size of data necessary for the statistical validity of a model; the higher the complexity and the lower the positive to negative ratio, the more data is required to train an appropriate model.

Albeit a lot of studies are already published in this area, there is yet to be any concrete headway commensurate with the amount of research effort so far. Obviously, there is a need for more collaborative effort among research teams. Public availability of data and process description need to be considered for convenient study reproduction. More efforts need to be channelled into tools packaged for cross-domain use.

Overall, the study finding shows that in respect of RQA, the SVM and ensemble method as the most used algorithms with no justification given in the majority of the cases as to why an algorithm is preferred to another neither did the studies explicitly discuss the effect the small sizes of the data could have on the models. In respect of RQB, the positive examples was found to be on the average less than 10% of the data sizes. The CV was found used heavily among the studies evaluated. This is an indication that the effect of the small size of the datasets and the skewed positive-negative examples might have been acknowledged even if not explicitly discussed. The recall, precision, F-measure and AUC remain the favourable metrics across the studies. In respect of RQC, the study finds that there is not much consistency among the studies to establish ground for a meaningful comparison of the studies' results.

3.8 Summary

The mapping study and supplementary review reported in this chapter explored the TM methods that are currently being proposed to reduce the time and effort expended on screening of citations in SRs. It also investigated the information provided on the methods and the effects of data size on classification decisions, including any justification given for the choice of the models.

A set of 44 papers included in an existing SR on the subject were adopted. After the study selection process, 36 papers were initially included for this study. 35 papers were finally included after one of the papers (unpublished) could not be located. An additional search for relevant literatures not covered in the SR (covering the period between February 2014 to January 2018) identified 11 more papers (see Section 3.5).

The results showed that the field is steadily growing with varying tools and methods being proposed. Support vector machine was found to be the predominant method. The use of active learning methods appeared to be attracting increasing interest in more recent studies to provide minimal label possible for effective classification. The BOW on the tf-idf representation was found to be the most popular way to represent features for classification. WSS was also found to be attracting growing attention as a metric for assessing classifier performance. On comparability of result, few studies were found comparing their results to one or two others, the research undertaken by A. M. Cohen et al. (2006) is being used by most as a benchmark. However, it was interesting to notice a lack of any actual reproduction or replication of studies. There was lack of technical information related to the models being used and why they were chosen. There were no information regarding experimental processes, model parameter settings and how the settings were determined.

This chapter has identified how research into investigating how the information contained in studies in this domain support reproducibility is required, how likely complex and statistically valid are the TM models being proposed for automatic CS and the possibility of a tool for automatic CS with sufficient details to support the independent reproduction of its results.

Other literature that had significant influence on this research, including those that are related to SRs in other domains such as healthcare and social science (see Section 1.1), TM and ML (see Sections 2.2 and 2.1), efforts at ensuring reproducibility of computational studies (see Section 4.1.1) and model selection considering complexity (see Section 5.2), are discussed elsewhere in this Thesis.

CHAPTER 4

Reproducibility Assessment

One of the findings from the review presented in Chapter 3 is a lack of sufficient information in the studies reporting automatic CS experiments. Particularly, information on the technical details of the models and the experiments' processes. A lack of replicated or reproduced studies by independent researchers was also identified. In this chapter, how well the information contained in the study reports support their reproducibility is further investigated. Identification of factors responsible for the reproducibility of automatic CS studies based on TM techniques will contribute to answering one of the research questions of this research. The investigation is approached in two forms, first is an attempt to actually reproduce six of the 35 studies included for the review in Chapter 3 and second is the development of an evaluation protocol combining the literature and experience from first exercise to assess 33 of the 35 studies (two were excluded) for reproducibility. The findings provide insight into the basic set of information required of an automatic CS study to enhance its reproducibility chances. A checklist is proposed based on the study results. The findings from this study have been reported as a journal article in (Olorisade et al., 2017c), as a workshop paper and a poster at the Reproducibility in Machine Learning workshop of the 34th International conference on machine learning, ICML'17 (Olorisade et al., 2017b).

4.1 Introduction

In Chapter 3, 45 studies were identified from 35 articles on TM based techniques for automatic screening of citations in SRs. This study found the information provided on the models and the study process in the articles to be inadequate particularly in revealing the quality of the models. The review also identified a lack of independent replication or reproduction among the studies. Though, it found some studies comparing their results to others and even sometimes adopting some of the methods for data preparation in the other study. Whilst this is promising in a relatively new

field of research like this, there was no evidence to support that they were (intended) replication or reproductions of the earlier studies.

The importance of reproducibility in an emerging field like this is summed in the words of Aarts et al. (2015), *innovation points out paths that are possible; replication points out paths that are likely; progress relies on both*. The study reported in this chapter addressed reproduction issues in the automatic CS studies by assessing how easy it is to reproduce the results published in 33 articles. The 33 articles are part of those selected for the review in Chapter 3. The reported study entailed three steps: a reproduction attempt of six studies, development of an assessment framework on experience-driven identification of elements of the TM process essential to reproduction and a systematic assessment of the 33 studies based using the framework. The assessment attempted to reveal the reproduction strength or otherwise of each study and identified the element(s) that contributed to it.

The rest of this chapter is organised as follows: The conduct of the reproduction analysis is the focus of Section 4.2.1. Section 4.2.2 presents the definition of the assessment framework by defining the TM process elements to us for assessing each article, the attributes of each element as well as values and tags to measure the attributes and summarise the measured values of the attributes. The application of the assessment framework is presented in Section 4.2.3. The results from these activities is presented in Sections 4.3.1, 4.3.2 and 4.3.3 with the limitations of the study results in Section 4.3.4. An update information on the datasets used in the study is presented in Section 4.4. This is followed by a discussion of the study in Section 4.5 and a summary in Section 4.6.

4.1.1 Reproduction of computational studies

The issue surrounding the ability of independent researchers to reproduce computational studies has been identified in the past few decades and researchers have made several proposals about how to make computational studies reproducible. Mesirov (2010), Davison (2012) advised cultivating reproducibility into a habit and everyday research culture before its effect can be successfully noticed in publications.

Explicit and unambiguous description of processes and results is the first step towards ensuring independent researchers can clearly understand a study to the level that it can be reproduced by them (Mesirov, 2010). Undocumented implicit knowledge is often the main impediment to the implementation of proposed algorithms and models (Crick, Hall, & Ishtiaq, 2014).

Technology can support reproducibility (Hothorn, Held, & Friede, 2009). For example, it has been suggested that researchers should utilize whenever they can, available libraries and packages that are easily accessible to the public, are robust and are continually maintained (Davison, 2012; Mesirov, 2010). Cross platform soft-

ware should be chosen where possible for experiment purposes (Crick et al., 2014; Davison, 2012). However, it is practically impossible to capture all the decisions and situations during a computational study, so employing an automatic means of storing the details of every decision, process and result is encouraged (R. D. Peng, 2011; Sandve, Nekrutenko, Taylor, & Hovig, 2013; Davison, 2012). GitHub and other similar version control applications can aid capturing of the different stages and changes in experiments as well as providing long term storage and access to the digital artefacts (R. D. Peng, 2011; Sandve et al., 2013; Davison, 2012).

Public repositories and publishers are helping to ensure digital components of publications are available to readers (Mesirov, 2010; R. D. Peng, 2011); however, this does not guarantee that a study will be reproducible. Understanding the provided files is key to making independent (active) use of them but data files are still formatted haphazardly; partially or insufficiently annotated (Ioannidis et al., 2009; Rung & Brazma, 2013); codes are poorly commented while graphs and charts are sparsely annotated amongst other issues (Wilkinson et al., 2016). Though, the digital components storage provision facilities is a step in the right direction.

In order to ensure reproducibility, comparability and generalizability of studies, the Information Retrieval (IR) community have dedicated considerable efforts (notably) to the standardization of data formats to facilitate uniform storage, access and exchange of data, as well as the creation of common evaluation methods for techniques (Comeau et al., 2013; Zobel, Webber, Sanderson, & Moffat, 2011). Notable initiatives that have pushed research achievements in IR are TREC[1], CLEF[2] and NII Testbeds and Community for Information access Research (NTCIR)[3] (Agosti et al., 2012; Freire, Fuhr, & Rauber, 2016; Ferro, 2017). These efforts are inherently beneficial to and directly utilized in TM research. Some of the experimental collections used in TM are part of the experimental collections from real domains of interest like medicine, made available through the efforts of IR research at ensuring reproducibility and comparability in the field (Zobel et al., 2011). An example is the TREC collection, one of which is the corpus used in this work and in studies reviewed in this work. The evaluation metrics proposed and used in IR research are also beneficial to and utilized by TM studies (Hersh, 2005).

The Big Data to Knowledge (BD2K), trans-National Institute of Health (NIH) initiative has been established to facilitate the standardization, discovery and reuse of digital assets in biomedical research through innovative approaches and tools so that machines without human intervention can automatically access and (re)use study data. This initiative led to the agreement on the Findable, Accessible, Interoperable and Reusable (FAIR) principles that should guide such big data driven research. The

[1]http://trec.nist.gov/
[2]http://www.clef-initiative.eu/
[3]http://research.nii.ac.jp/ntcir

guidelines for these principles are described in (Wilkinson et al., 2016) and a sample tool implementation is provided in (Wilkinson et al., 2017).

These principles along with other aims of the BD2K initiative[4] support reproducibility of experiments by facilitating digital assets discovery (open knowledge) for verification, knowledge advancement and community wide research engagement. The realization of the BD2K objectives will not only be useful to biomedical research but also for the general science communities' effort on reproducibility of scientific research.

Data format is also key to access and reuse. Researchers should attempt to store their data in common formats (Crick et al., 2014) like the comma separated values (csv) or similar formats. This way, other researchers will find it easier to retrieve and manipulate the data.

Prior to publication, it has been suggested that researchers should conduct a reproducibility check by asking colleagues not involved in the research to attempt to reproduce their studies based strictly on the information contained in their manuscript. This way, it will be possible to anticipate areas of ambiguities and insufficient information (Ioannidis et al., 2009; Mesirov, 2010; Sandve et al., 2013).

Though reproducibility is not a license to a study's correctness, validity or quality, it is however, a precursor to these qualities as utilizing these principles will not only aid the reproducibility of studies but also further the development of the means to ensure it.

4.2 Study reproducibility

The conduct of the study is discussed in this section. The study involved three steps:

i) Reproduction analysis: this step is an attempt to reproduce the results for six studies. The six studies were selected because they were shared a common dataset.

ii) Assessment framework definition: this step involved the identification of relevant assessable elements of the TM process, definition of values to measure the elements and tags to summarise the assessment.

iii) Reproducibility assessment: this step involved the application of the assessment framework in step ii to each of the studies.

The objective of this study was to investigate how reproducible are the results of the TM studies on automatic CS. This is achieved by attempting to replicate some of the studies, taking note of the factors that contributed to the success or otherwise of 'their reproducibility' and using the experience to methodically evaluate the rest.

[4]https://commonfund.nih.gov/bd2k

Reproducibility refers to the ability to independently reproduce the results reported in each of the studies by replicating the experimental steps based exclusively on the information provided in the reports.

4.2.1 Reproduction analysis

The experience during the attempt to reproduce six of the experiments is discussed in this section. For the reproduction analysis, six studies (A. M. Cohen et al., 2006; Matwin et al., 2010; Kim & Choi, 2012; A. M. Cohen et al., 2012, 2009, 2010) based on various topics in the Drug Evaluation Review Program (DERP) datasets were selected, particularly the topics contained in the Text REtrieval Conference (TREC) 2004 Genomics track corpus[5] (A. M. Cohen & Yen, 2014). The datasets are part of the DERP reports made available through the collaboration between the Cochrane Centre and the Evidence-based Practice Center (EPC) of the Agency for Healthcare Research and Quality (AHRQ) (A. M. Cohen et al., 2006). From now on, this dataset will referred to as the DERP dataset.

4.2.1.1 Data retrieval

Despite the prevalence of the datasets among the studies, none of the studies made the subset used for their studies available for reuse. The closest reusable information was the PMID information and the link to the raw dataset provided in (A. M. Cohen et al., 2006) which was found to be obsolete. The updated link[5] as at the time this study was undertaken has also become invalid as at the time of writing this thesis [6].

The raw dataset contained 50 reviews of 4,367,228 articles, separated into a few files in two major formats - eXtended Markup Language (XML) and Standard Generalized Markup Language (SGML). The subset required for this study was 15 reviews of 18,733 articles. The documentation for the datasets was provided in 10 files which had to be studied to understand the structure and content of the data files. A parser was subsequently developed to retrieve the datasets of interest from the whole set using the Pubmed Identification (PMID) provided by A. M. Cohen et al. (2006).

4.2.1.2 Preprocessing

It was fairly possible to replicate the text pre-processing steps reported in the studies. Though, there were no intermediate results provided in any of the studies reports that could be compared to except (A. M. Cohen et al., 2006). Some of the studies also adopted the preprocessing as described in (A. M. Cohen et al., 2006). The preprocessing activities (see Section 2.2.2 for more details on these steps in TM)

[5] http://skynet.ohsu.edu/trec-gen/data/2004/

[6]. The datasets are currently hosted at https://dmice.ohsu.edu/trec-gen/data/2004/

involved items (ii) – (v) as enumerated below and discussed in Section 2.2. Item (i) was an activity specific to (A. M. Cohen et al., 2006).

i) appending special tags to features from the Medical Subject Heading (MeSH) and publication type before the preprocessing steps as was done in (A. M. Cohen et al., 2006; Matwin et al., 2010).

ii) stopwords removal – removing commonly used words (e.g. articles and prepositions) referred to as stopwords.

iii) tokenization – breaking the sentences into words or phrases known as features.

iv) feature representation – representing or encoding the features in a numeric usually binary or frequency based - format.

v) storing all tokens in a feature vector using the BOW approach.

The preprocessing steps in A. M. Cohen et al. (2006) were followed for this study since they provided the most explicit discussion with intermediate results. Accordingly, a distinction was made between three types of features from the corpus – title and abstract, the MeSH terms and publication type. These MeSH terms were appended with 'mh' while the publication type were appended with 'pt' to distinguish them from similar title and abstract terms. These terms were appended before removing any stopwords. Cohen et al.'s article was not clear on whether the appending was performed before or after the removal of stopwords. The features were represented in a binary format in a BOW.

4.2.1.3 Feature selection

The feature selection process as presented in Section 2.2.4 is part of the methodologies used to reduce the dimension of the feature vector without losing any important information, and often lead to improved performance of the predict models. In this study, the χ^2 method (see Section 2.2.4) was used according to the process reported in (A. M. Cohen et al., 2006). The RapidMiner data science platform[7] and the FSelector [8] (version 0.21) package in 'R' for feature selection were employed for this purpose.

4.2.1.4 Model training

New algorithms were proposed in (A. M. Cohen et al., 2006; Matwin et al., 2010; A. M. Cohen et al., 2009) but no implementation (or reference to it) were provided for the algorithms they proposed, therefore, the base algorithms in each case were used for the purpose of the experiment. The experiments were conducted with the

[7]https://rapidminer.com/
[8]https://cran.r-project.org/web/packages/FSelector/index.html

simple Perceptron and SVM algorithms in Python's 'sklearn' package (Pedregosa et al., 2011) and the implementation of the 'votedperceptron' algorithm provided in Weka (with no weighting) (M. Hall et al., 2009), which is the algorithm that was modified in (A. M. Cohen et al., 2006).

The data from the different reviews in the dataset were kept and used in the order maintained in the file provided by Cohen et al. [5]. The supporting materials – codes and data files – to aid the reproduction of this study was hosted on github[9]. The parameters set for the different classifiers built from the SVM and the perceptron algorithms are presented in Tables B.1 and B.2 respectively.

In both cases, the sample weight for the negative class to positive class was set at 1:4 during training. This sample weighting was chosen following (A. M. Cohen et al., 2006), which used the same weight for some of the studies reported there. The best performance for each of the fifteen studies was recorded at different weights in (A. M. Cohen et al., 2006), the weighting of 1:4 showed a consistent acceptable performance comparable to the best cases for all studies (in some cases providing the best results) (A. M. Cohen et al., 2006). Thus, a consistent value ratio of 1:4 that generally performed well was chosen for the purpose of experimentation.

The studies mostly used the cross validation approach during model training. This might be due to the (relative) small size of the datasets. The 'StratifiedKFold' method from Python's 'sklearn' package was used to divide the datasets into training and test data for the 5x2-fold CV. The method ensured maintaining the negative:positive class ratio in both the training and test data as was in the original dataset. The random_state parameter of the method was set to a value of '67' (chosen at random), during the partitioning for the cross validation. The random_state is the seed of the pseudo-random number generator used when shuffling the data. The shuffling ensured that each run of the algorithm produced different results.

4.2.1.5 Model assessment

The average precision, recall and F_1 scores were calculated using the precision_score, recal_score and F_1_score methods in 'sklearn'. The average parameter in these methods was set to binary corresponding to a binary classification. Refer to Section 2.3 for brief definition of the metrics.

4.2.2 Assessment framework definition

The approach proposed by (González-Barahona & Robles, 2012) was adopted and modified to suit this study's need in line with the TM process described in Section 2.2 and depicted in Figure 2.4. González-Barahona and Robles (2012) identified a set of information elements required to support the reproducibility of SE studies based

[9]https://github.com/raylite/reproducibility-data

Table 4.1: Values describing the attributes

S/N	Attribute	Values
1	Identification	Complete (Classical), Partial, No, N/A
2	Description	Complete (Textual), Partial, No, N/A
3	Availability	Private, Public (Free), No, N/A
4	Persistence	Likely, Unknown, N/A
5	Flexibility	Complete, Partial, No, N/A

on data. The elements were: data source, retrieval methodology, raw dataset, study parameters, extraction methodology, processed data, analysis methodology and results dataset (González-Barahona & Robles, 2012). Their proposal built on the Knowledge Discovery in Databases (KDD) schema proposed by Fayyad, Piatetsky-Shapiro, and Smyth (1996) where data, selection, target data, preprocessing, preprocessed data, transformation, transformed data, data mining, patterns, interpretation/evaluation and knowledge were identified as the elements composing the KDD process.

In their study, González-Barahona and Robles (2012) defined five attributes and some possible values that can be assigned to the attributes. The five attributes are:

i) Identification (location): where the information element can be accessed e.g. web-link.

ii) Description: level of published details provided about the information element including it's internal organization and structure, and its semantics.

iii) Availability: a measure of the difficulty involved to currently access or acquire the information element.

iv) Persistence: the possibility of the information element being available for future use.

v) Flexibility: how adaptable is the information element to different formats and/or environments.

These attributes can be assessed independently of each other based on the available information in a publication. The values that can be assigned to each attribute are listed in Table 4.1.

The values in bracket are possible replacements used when most appropriate in the context of this study. Further details on the adaptation of the elements as suited to this study are presented later in this section.

González-Barahona and Robles (2012) also defined a set of six (summary) assessment tags (Table 4.2) that may be combined, as applicable, to summarize the

strength or otherwise of the contribution of an element to the reproducibility of a study.

Table 4.2: Summary assessment tags

S/N	Tag	Meaning
1	U	Usable for reproduction
2	D	Usable for reproduction with some difficulty
3	N	Not usable for reproduction
4	+	Future availability is foreseeable
5	*	Flexible
6	-	Irrelevant

In accordance with the procedure of González-Barahona and Robles (2012), the identified elements for this study are presented in Section 4.3.2. The relationship between the identified elements in the TM process is depicted in Figure 4.1. The difference between this process and Robles et al.'s proposal are:

- the data source element was added to capture data retrieval
- the interpretation/evaluation step is replaced with model assessment

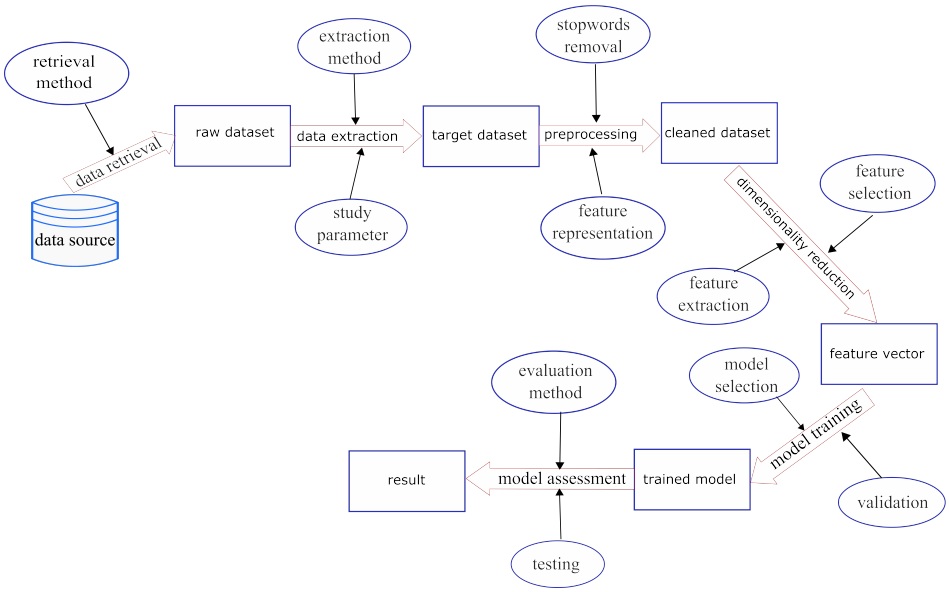

Figure 4.1: Detailed TM process with intermediate output

These elements were assessed using the attributes and values (with minor adaptation) as proposed by González-Barahona and Robles (2012) (see Tables 4.3 and 4.1). The interpretation of the values varies according to the attribute-element context. The meanings as used in this study are provided below:

- Complete: This value was associated with three attributes – identification, description and flexibility. It implies that basic information needed to locate or identify the element in question based on the attribute was provided. For example, in the case of raw datasets, this may imply the general name of a particular dataset with the associated link from where it can be retrieved. Notable variations are:

  - Classical: the term 'classical' was sometimes used (instead of complete) under identification, if one of the traditional ML algorithms was used out of the box with no (significant) alteration. This term was preferred to indicate that insufficient description may be tolerated in such cases.
  - Textual: 'Textual' was used (under description) to indicate a new method, tool or algorithm proposed by the researchers and described only with text in the publication with neither source code nor executable file provided.

- Partial: This value was associated with identification, description and flexibility attributes. It was was used to indicate situations where the information provided about an element was too general or insufficient to be reproduced. For example, a dataset (source) named with no link information to its exact webpage but rather to the index page of the provider where the researcher will be left to try and navigate to the desired resource.
- No: No implies complete absence of an attribute.
- N/A (Not Applicable): This implies the attribute was not applicable to the element in question. For example, for a study that did not make use of any of the information elements described above, the corresponding entries will be N/A.
- Likely: This value applies to the persistence attribute if there was a possibility that a relevant element is likely to be available for future access.
- Private/Public/Free: These terms were associated with the availability attribute. The term private was used to indicate elements, in this case data or tools, located but inaccessible due to extra constraints like membership, application, subscription etc. imposed before access may be granted. 'Public' on the other hand meant that the dataset (raw or processed) was provided for public use. 'Free' was used in the case of a tool used that is available for free download.
- Unknown: This value was used in association with the 'availability' attribute, when it was not easy to determine whether or not a relevant element will be available for future access.

Not all attributes were defined for every element. Presented in Table 4.3 is an example of the set of attributes applicable to each element.

Table 4.3: Attributes-element combination

	Data sources	Datasets	Technique	Parameters	Tools/Algorithms
Identification	•	•	•	•	•
Description	•	•	•	•	•
Availability	•	•			•
Persistence	•	•			•
Flexibility		•			•

4.2.3 Reproducibility assessment

After the attempt to reproduce the experiments reported in the six papers, the experience provided a solid basis for the evaluation of how easy it might be to reproduce the rest of the studies and to identify what factors might influence the extent of their reproducibility. In each study, the different information elements (depicted in Figure 4.1 and explained in Section 4.3.2) were identified, then the assessment attributes and their associated measuring values highlighted in Section 4.2.2 were used to indicate a measure of the extent of usefulness (in supporting reproduction) of the provided information. The assessment framework was applied to 33 studies.

4.3 Results

In this section, the results of the three basic activities of this study described in Section 4.2 are presented.

4.3.1 Reproduction analysis

The outcomes from the reproduction analysis of the six studies (described in Section 4.2.1) are reported in this section. The difficulties encountered were very similar across all of the studies. Nevertheless, when there is a need to show a concrete example, (A. M. Cohen et al., 2006) which provides the most detailed step by step measurable outputs will be referred to.

Generally, it was difficult to acquire the raw/cleaned dataset used in the studies. Often the referenced web links were either broken or pointed to the index page of the hosting institution. In most cases, however, there was no link even to the location of the raw dataset. The papers contained sufficient information that identified the classification algorithm used but the provided information was not effective to reproduce the classification results. Beyond the standard algorithms, all the studies attempted something new to try to optimize the performance of the traditional al-

gorithms and mitigate the effect of any known (citation) text classification problems like class imbalance. However, they provided only textual descriptions of the changes or at best an algorithm of the changes but not the code that was used.

Starting from the dataset, an analysis of the details available in each of the studies are as below:

- The link to supplementary materials provided in (A. M. Cohen et al., 2006) was broken. The new link was located but the cleaned extracted dataset was not provided. The site contained a web link to the TREC 2004 Genomics Track webpage but not directly to where the raw data was supposed to be located; the direct link[5] that was found during the course of this study. As at the time of writing this thesis, this link has also become invalid with the new link provided[6]. They also provided a file with the PMIDs for the datasets they used.
- (Matwin et al., 2010) referenced the information provided in (A. M. Cohen et al., 2006).
- Dataset source or location was not provided in (Kim & Choi, 2012).
- Data source or body providing the data was named in (A. M. Cohen et al., 2012, 2009, 2010) but neither a link nor any other retrieval information was provided for the dataset used.

The complete dataset based on the PMID index file made available by A. M. Cohen et al. (2006) could not be retrieved from the OHSU repository for the TREC 2004 datasets. 18,431 data files were retrieved from the directories: "2004_TREC_ASCII MEDLINE_1" and "2004_TREC_ASCII_MEDLINE_2" out of the 18,733 data files that made up the 15 reviews used in (A. M. Cohen et al., 2006; Kim & Choi, 2012; Matwin et al., 2010). The 302 missing files (see Table 4.4 for the number of studies retrieved for each topic) could not be located. Thus, the reproduction analysis relied on an incomplete dataset, which was a significant setback from the perspective of reproducibility. In order to circumvent this problem, the corresponding author of (A. M. Cohen et al., 2006) was contacted requesting the extracted dataset used in their experiment and stated our mission but got no response. (A. M. Cohen et al., 2012, 2009, 2010) used part or all of this dataset and also additional data.

The information provided about pre-processing and feature selection was mostly useful for reproduction across the papers. The feature representation used was only reported in (A. M. Cohen et al., 2012). There was no explicit explanation of the representation. In (A. M. Cohen et al., 2006), the paper describes how they selected statistically significant features using the χ^2 method with 0.05 α level, thus it was easy to compare results. The two applications used (Rapidminer and Fselector) provided consistent results for more than top 50% of the selected features and above 80% in total for all the selected features. Despite this, it was impossible to produce

Table 4.4: Retrieved corpus size(s) and number of top features $\alpha = 0.05$ (Cohen et al.'s appears in italics)

Review topics	Corpus	χ^2 top features	MeSH features	PubType features
ACEInhibitors	2498	242	54	7
	2544	*210*	*40*	*5*
ADHD	845	115	39	0
	851	*80*	*24*	*0*
Antihistamines	308	31	10	1
	310	*29*	*9*	*0*
AtypicalAntipsychotics	1115	173	44	7
	1120	*302*	*71*	*8*
BetaBlockers	2043	129	26	3
	2072	*194*	*42*	*5*
CalciumChannelBlockers	1190	166	43	4
	1218	*329*	*77*	*5*
Estrogens	362	102	26	4
	368	*233*	*44*	*5*
NSAIDs	389	146	39	5
	393	*242*	*51*	*5*
Opioids	1883	78	25	0
	1915	*55*	*14*	*0*
OralHypoglycemics	493	97	22	3
	503	*234*	*55*	*4*
ProtonPumpInhibitors	1314	165	40	4
	1333	*206*	*54*	*6*
SkeletalMuscleRelaxants	1610	67	14	4
	1643	*11*	*2*	*2*
Statins	3402	173	39	5
	3465	*467*	*87*	*6*
Triptans	657	226	42	6
	675	*121*	*22*	*3*
UrinaryIncontinence	322	137	37	6
	327	*215*	*45*	*5*

the exact number of features for a 0.05 confidence interval using the χ^2 method with Cohen et al.'s. This might have been caused by the lack of complete dataset in the first place. Another possibility is that there might have been some fine-tuning steps not reported in the paper because the discrepancy in the number of features found in this study and theirs was too wide in some cases. The results of the data retrieval and feature selection are presented in Table 4.4 alongside the results (in italics) of same exercise by A. M. Cohen et al. (2006).

The 5x2-fold CV average results for precision, recall and harmonic mean (F1) are presented in Table 4.5, alongside an extract from Cohen et al.'s results (A. M. Cohen et al., 2006) in italics. The results were based on the number of top features reported in (A. M. Cohen et al., 2006) rather than the study's result from applying the χ^2 method at $alpha = 0.05$. The 'votedperceptron' precision values from this study were better than Cohen et. al.'s but the recall and F1 scores were worse. The lower recall in this case accounted for the higher precision values, since there is always a trade-off between recall and precision. But the simple perceptron and SVM showed comparable and sometimes lower recall with higher precision performance compared to Cohen et al.'s. This showed that the results of the studies could be reproduced only if the authors were to provide sufficient information on experimental procedure and data. If we had access to the full dataset, it might still be impossible for us to get the exact classification outcomes given that randomization is usually involved in the procedures of text classification algorithms and none of the papers provided access to neither the data partition nor the indices they used for the training and test/validation sets. They only provided proportion information about training and test sets (i.e. what percentage of the data was used for these purposes). The seed value used (if any), would have been sufficient to reproduce any randomised step but that was not provided either.

Table 4.5: 5X2-fold CV results

Review topics	Method	Precision	Recall	F1
ACEInhibitors	Cohen	*0.0387*	*0.9561*	*0.0745*
	Votedperceptron	0.414	0.101	0.16
	Simple perceptron	0.11	0.86	0.19
	SVM	0.15	0.75	0.25
ADHD	Cohen	*0.0945*	*0.9200*	*0.1713*
	Votedperceptron	0.53	0.514	0.521
	Simple perceptron	0.35	0.95	0.50
	SVM	0.46	0.94	0.62
Antihistamines	Cohen	*0.0502*	*0.8500*	*0.0948*
	Votedperceptron	0.571	0.467	0.517

Table 4.5: 5X2-fold CV results (continued)

	Simple perceptron	0.40	0.83	0.53
	SVM	0.40	0.98	0.57
AtypicalAntipsychotics	Cohen	*0.1534*	*0.9493*	*0.2642*
	Votedperceptron	0.582	0.533	0.556
	Simple perceptron	0.42	0.80	0.53
	SVM	0.33	1.00	0.49
BetaBlockers	Cohen	*0.0334*	*0.9286*	*0.0644*
	Votedperceptron	0.459	0.201	0.279
	Simple perceptron	0.19	0.85	0.31
	SVM	0.18	0.97	0.30
CalciumChannelBlockers	Cohen	*0.0952*	*0.9460*	*0.1730*
	Votedperceptron	0.581	0.447	0.503
	Simple perceptron	0.38	0.78	0.49
	SVM	0.41	0.97	0.26
Estrogens	Cohen	*0.2252*	*0.9725*	*0.4044*
	Votedperceptron	0.645	0.440	0.519
	Simple perceptron	0.32	0.83	0.44
	SVM	0.38	0.96	0.54
NSAIDs	Cohen	*0.2631*	*0.9317*	*0.4103*
	Votedperceptron	0.651	0.568	0.603
	Simple perceptron	0.36	0.95	0.51
	SVM	0.44	0.92	0.59
Opioids	Cohen	*0.0092*	*0.9467*	*0.0182*
	Votedperceptron	0.359	0.068	0.114
	Simple perceptron	0.04	0.84	0.07
	SVM	0.08	0.56	0.14
OralHypoglycemics	Cohen	*0.4004*	*0.9471*	*0.4561*
	Votedperceptron	0.35	0.86	0.49
	Simple perceptron	0.67	0.75	0.68
	SVM	0.28	1.00	0.44
ProtonPumpInhibitors	Cohen	*0.0602*	*0.9373*	*0.1132*
	Votedperceptron	0.519	0.301	0.380
	Simple perceptron	0.26	0.80	0.38
	SVM	0.24	0.93	0.38
SkeletalMuscleRelaxants	Cohen	*0.0055*	*1.0000*	*0.0109*
	Votedperceptron	0.428	0.067	0.120
	Simple perceptron	0.03	0.94	0.05
	SVM	0.04	0.67	0.08

Table 4.5: 5X2-fold CV results (continued)

Statins	Cohen	*0.0311*	*0.9647*	*0.0603*
	Votedperceptron	0.272	0.039	0.070
	Simple perceptron	0.07	0.87	0.12
	SVM	0.11	0.69	0.19
Triptans	Cohen	*0.0365*	*0.9583*	*0.0703*
	Votedperceptron	0.647	0.634	0.641
	Simple perceptron	0.45	0.92	0.82
	SVM	0.48	0.93	0.63
UrinaryIncontinence	Cohen	*0.1559*	*0.9850*	*0.2691*
	Votedperceptron	0.473	0.465	0.465
	Simple perceptron	0.33	0.84	0.46
	SVM	0.26	0.97	0.41

Overall, based on the reproduction analysis experience, it can be concluded that it is difficult to reproduce the studies. This difficulty could have been significantly reduced if the studies had made available the datasets they used, the seed value for each of the randomisation steps or the data partition or indices for the training and test/validation sets, and the implementation details of any algorithm or method used.

4.3.2 Definition of the assessment framework

The following information elements were identified as being required to support the reproducibility of TM application in the context of CS following the attempt to reproduce results of the six studies:

i) Data source: The actual location of the raw dataset – direct web-page.

ii) Raw data: The whole of the dataset retrievable from (i). Necessary information may include the description of the internal structure of the dataset, the retrieval method, the file format(s) etc.

iii) Dataset: The focused dataset used in a particular TM experiment which may be the whole of (ii) or a subset. Information required may involve any new location of the extracted dataset, the extraction technique and the portion extracted.

iv) Pre-processing: This involved preprocessing steps of tokenization and noise removal from the resulting dataset.

v) Feature representation: The method used for numerical encoding of the features.

vi) Feature Selection: The feature selection/reduction approach used with sufficient details.

vii) Dimensionality reduction: Any other method used to further reduce the dimensionality of the feature vector beside feature selection.

viii) Data partitions: Partitions (or indices) of the data used for the different classification operations – training, testing and or validation or seed value used to control randomised partitioning.

ix) Modelling: Details of the ML algorithm used for mining the text, seed values for randomisation control, algorithm parameters and code or executable file for newly proposed algorithms.

x) Model assessment: The testing or validation approach used.

xi) Third party framework: Available ML software or packages used during the experiments.

xii) Custom method: This referred to algorithms or techniques proposed by the authors in a study.

4.3.3 Reproducibility assessment

Based on the information elements, attributes, values and tags defined in Sections 4.3.2 and 4.2.2, the reproducibility of 33 studies on the application of TM to CS in SRs were assessed. A typical detailed assessment of a study is shown in Table 4.6 while the overall assessment is presented in Table 4.7.

The issues related to data sources and datasets posed a key challenge to reproduction as information found in 28 (85%) of the papers (in both elements) are only useful with some difficulty while four (12%) were found not to have useful information about the data source and six (18%) about the dataset. 13 (39%), 16 (48%) and 11 (33%) of the papers respectively provided pre-processing, feature selection and dimensionality reduction information that was fully useful to reproduction; an additional six (18%), eight (24%) and four (12%) respectively with some difficulty. This implied an average of five (15%) with either irrelevant or not useful information. Pre-processing and feature representation recorded values higher than 30% on no useful information mainly because the authors might have assumed implicit understanding thereby failing to mention what steps were specifically taken in data cleaning e.g. were stopwords removed. This information is necessary because there have been situations where experiments were conducted with stopwords. In the terms of data split, only five (15%) articles provided information that may be useful for reproduction. The information about the ML algorithms could be used for reproduction in nine (27%) articles and with difficulty in another 19 (57%). However, information provided on custom (proposed) methods in three (9%) articles were

Table 4.6: A typical assessment output of a study (see footnote for abbreviations in column 1)

Elements	Identification	Description	Availability	Persistence	Flexibility	Assessment
DS	Partial	No	Private	Likely	N/A	D+
Dataset	No	No	Unknown	Unknown	No	N
PP	Classical	Complete	N/A	N/A	N/A	U
FS	Classical	Complete	N/A	N/A	N/A	U
DR	Classical	Complete	N/A	N/A	N/A	U
Split	No	Partial	N/A	N/A	N/A	D
Technique	Classical	Partial	N/A	N/A	N/A	D
Testing	Complete	Partial	N/A	N/A	N/A	D
TPF	Complete	Complete	Free	Likely	No	U+
CM	N/A	N/A	N/A	N/A	N/A	—

Note: DS – Data source; PP – Pre-processing; FS – Feature selection; DR – Dimensionality reduction; Split – dataset partition; Technique – Modelling technique/algorithm used; Testing – testing or cross validation technique; TPF – Third party framework and CM – Custom method.

found to be useful, 16 (48%) with difficulty while 13 (39%) had no provision for this element.

Validation and testing information were found useful for reproduction in 13 (39%) of the articles and with some level of difficulty in 12 (36%). Finally, all third party tools or frameworks used in the studies were found to be free and accessible. The information provided on them was sufficient to locate the tools.

4.3.4 Threats to study validity

Certain threats limiting the validity of the study in three ways can be identified. The first affects the internal validity originating from the subjective nature of the defined elements and assessment process. Each study was assessed based on the understanding of the information provided in the papers. The defined elements are however based on experience and previous research that has defined elements for similar processes. The elements were defined as a general framework to cover the TM process and is subject to future refinement. It however suffices for the purpose of illustrating reproducibility in this study. The attribute values and summary tags are also categorical metrics and are subjectively assessed. However, as independent assessors, the evaluator had no vested interest in any of the candidate studies neither was the work of any of the study team members involved.

The second type of validity threat relates to construct validity stemming from

Table 4.7: Summary assessment of the reproducibility assessment

Paper	DS	Dataset	PP	Feature rep.	DR	Split	Technique	Validation/ Testing	TPF	CM/ Tool	Ref
P1	D+	N	U	U	U	D	D	D	U+	-	Bekhuis and Demner-Fushman (2010)
P2	D	D	U	U	U	N	U	D	U	D	Bekhuis and Demner-Fushman (2012)
P3	D+	D	U	U	U	N	U	U	U+	D	Bekhuis, Tseytlin, Mitchell, and Demner-Fushman (2014)
P4	D+	D+	U	U	U	N	U	U	U+	D	S. Choi, Ryu, Yoo, and Choi (2012)
P5	D	D	U	U	D	N	D	D	-	D	(A. M. Cohen, Hersh, Peterson, & Yen, 2006)
P6	D+	D+*	U	U	U	N	D	D	-	D	A. M. Cohen (2006)
P8	D	D+	D	U	U	N	U	U	U+	-	A. M. Cohen (2008)
P9	D	D+	D	U	U	N	D	U	-	D	(A. M. Cohen, Ambert, & McDonagh, 2009)
P10	D+	D+*	N	N	N	N	U	U	-	-	A. M. Cohen, Ambert, and McDonagh (2012)
P11	D+	D+	N	N	D	N	U	U	-	-	Dalal et al. (2013)
P12	D	D+	N	U	N	N	U	U	-	-	A. M. Cohen, Ambert, and McDonagh (2010)
P13	N	N	-	-	-	-	-	-	U	U	Katia R Felizardo, Andery, Paulovich, Minghim, and Maldonado (2012)
P14	N	N	-	-	-	-	-	-	U	-	Katia R Felizardo et al. (2011)
P15	N	N	-	-	-	-	-	-	U	-	Katia Romero Felizardo, Souza, and Maldonado (2013)
P18	D+	N	D	U	U	D	U	D	U+	D	Frunza, Inkpen, and Matwin (2010)
P19	D	D	U	U	U	N	D	D	U+	D	Frunza, Inkpen, Matwin, Klement, and Oblenis (2011)
P20	N	D	U	U	U	N	U	D	U+*	-	Adeva, Atxa, Carrillo, and Zengotitabengoa (2014)
P21	D	D+	N	N	N	N	-	-	-	D	Jonnalagadda and Petitti (2013)
P22	D+	D+	U	D	N	N	D	D	-	-	Kim and Choi (2012)
P23	D	D	N	U	N	N	D	U	U+	N	Kouznetsov and Japkowicz (2010)
P24	D	D	N	D	D	D	D	U	-	D	Kouznetsov et al. (2009)
P26	D	N	-	-	-	-	-	-	U	-	Malheiros, Hohn, Pinho, and Mendonca (2007)
P27	U	D+*	N	D	N	-	D	-	U	-	Martinez, Karimi, Cavedon, and Baldwin (2008)
P29	D+	D	U	U	N	N	D	U	-	D	Matwin, Kouznetsov, Inkpen, Frunza, and O'blenis (2010)
P30	D	D	D	D	N	D	D	D	U+*	D	Miwa, Thomas, OMara-Eves, and Ananiadou (2014)
P32	D+	D	N	N	N	N	D	D	-	-	Shemilt et al. (2014)
P34	D+	D+	D	D	N	N	D	N	U	D	Tomassetti et al. (2011)
P37	D+	D+	N	D	N	N	D	D	-	D	Small, Wallace, Trikalinos, and Brodley (2011)
P38	D+	D	U	U	N	N	D	D	-	-	Wallace, Small, Brodley, Lau, Schmid, et al. (2012)
P40	D	D	N	N	-	-	D	U	-	U	Wallace, Small, Brodley, Lau, and Trikalinos (2010)
P41	D+	D	D	D	D	N	D	-	-	D	Wallace, Small, Brodley, and Trikalinos (2010)
P42	D	D	U	D	-	-	D	U	-	U	Wallace, Trikalinos, Lau, Brodley, and Schmid (2010)
P43	D+	D	U	U	U	N	D	U	-	D	W. Yu et al. (2008)

Note: U = Usable for reproduction; D = Usable for reproduction with some difficulty; N = Not usable for reproduction; + = Future availability is foreseeable; * = Flexible; - = Irrelevant

the studies involved in the assessment which were chosen based on a SR published in 2015 and thus may not represent the whole research area particularly studies published after the review was conducted.

The last threat imposes a limitation on the conclusion validity based on the fact that the assessment was conducted by a researcher with no other corroboration to cross-check correctness and validate judgements. Multiple assessors would have been preferable. Nevertheless, the effort was purely for academic and research progress with no preference or bias for/against any of the studies or authors of the studies assessed. So, while this threat was acknowledged, multiple assessors might not have had any significant impact on the outcome.

4.4 Data retrieval update

The EPC datasets retrieved for this study formed the bulk of the datasets for other studies reported in this thesis. Therefore, after the conclusion of this study, efforts continued at locating the full sets of data for the 15 reviews of the TREC 2004 datasets used in (A. M. Cohen et al., 2006). The complete datasets were later successfully retrieved and used in subsequent work. Further details on how the full set was retrieved is presented in Section 5.4.1.

4.5 Discussion

It was not possible to reproduce any of the results of the original studies during the reproduction analysis because the complete dataset could not be retrieved and, for all six studies, critical data usage information were missing. In particular, more information was needed about how the dataset was partitioned and about the seed values used for randomization.

Some of the papers assessed for reproducibility (e.g. P1 – P9, as shown in Table 4.7) did exhibit some potential for reproducibility providing good accessibility to raw datasets and useful explanations of their preprocessing, feature representation and dimensionality reduction processes. However, for many of the papers, information about dataset partitioning was inadequate.

In addition, access to and information about the dataset used and details about the algorithms used in the studies were insufficient for reproduction. In particular, information about parameters and new (proposed) algorithms was lacking.

Generally, the accessibility of third party tools was good but there was no assurance about their persistence and flexibility.

As a result of this study, a checklist (Table 4.8) based on the information elements identified in Section 4.3.2 was proposed.

The reproduction analysis and reproducibility assessment in this study revealed that the studies were hard to reproduce due to missing information regarding access to and availability of raw, target or processed datasets. Reproduction by independent research teams was possible but with different levels of difficulty specific to each study.

Studies in this field need to be reported with more information than is currently the practice, to aid independent reproduction of the studies. One possibility would be to create a common repository where research results can be stored along with associated datasets, partition information and process details (Gentleman et al., 2004). This would ensure persistence and availability of datasets, as well as providing additional experiment information not included in publications. Making the full code used during experiments available is advised. Also, communication may improve between researchers due to the need for further explanation or elicitation of undocumented tacit knowledge or ideas used in the original experiment (F. Shull et al., 2004; Vegas et al., 2006).

Data and process descriptions need to be made publicly available in order to support study reproduction and consequently enhance external validation and maturity chances of claims and discoveries. It will also help improve the availability of evidence about the effectiveness of the methods that have been proposed for the application of TM techniques to CS in SRs.

4.5.1 Reproducibility checklist

The findings in this study lead to the suggestion of a checklist (with the preliminary version (version 1.1) presented in Table 4.8) to support the reproducibility of TM studies and CS automation studies in particular. The checklist was built on the TM elements defined in Section 4.3.2. Unlike the mainstream ML studies on image classification where some benchmark datasets have been standardized and are easily retrievable through ML packages like 'keras' (Chollet, 2015), text data (e.g. SR datasets) still exist in various forms and repositories. Efforts of initiatives like the TREC[10] in the information retrieval domain are commendable and have been helpful at making shared corpora available for TM research. Therefore, there is the need to distinguish between the type of dataset information provided in a study, whether it was the raw data or the actual subset ((target dataset), if only part of a larger set) was used in a study.

The checklist can be used by authors reporting TM experiments for CS in SR or any text classification experiment to help improve reproducibility.

Reviewers may also use the checklist to assess the level of reproducibility of TM studies in the context of CS for SRs. It is expected that the checklist will continue to

[10]http://trec.nist.gov/

Table 4.8: The preliminary draft of the reproducibility enabling information checklist for TM studies - version 1.1

Item No.	Information elements	Yes	No	N/A
1	Original location of the raw dataset			
2	Provided link to local copy of: a. Raw dataset b. Target dataset c. Cleaned dataset			
3	Described the internal structure of: a. Raw dataset b. Target dataset c. Cleaned dataset			
4	Data retrieval method details			
5	Data extraction method described			
6	Pre-processing details			
7	Feature representation technique			
8	Feature selection technique			
9	Dimensionality reduction technique			
10	Final feature vector download link			
11	Training algorithm			
12	Custom algorithm a. Text b. Code c. Algorithm d. Executable file			
13	Model assessment method			
14	Detailed model assessment result			
15	Necessary seed values provided			
16	Training/test data partition available or indices provided a. Link to data partitions provided b. (link to) Indices provided c. Seed value provided			
17	Provide name and version number of third party or external software package used			

be evaluated and refined by other researchers. The checklist is in partial compliance with the FAIR principle as described in (Wilkinson et al., 2017). The data source and storage details will ensure the data is Findable, while being hosted on the internet at a published address will ensure it is Accessible. Interoperability is still a challenge, given that the data is being stored in popular formats on general-purpose repositories making it usable by humans, but not automatically usable by machines. The information about data format and partitioning will facilitate the Reusability of the data.

4.5.2 Checklist application

In a separate activity, the checklist (version 1.1) (Table 4.8) was revised, particularly in terms of the brevity and clarity of its wordings. In the revision, some items later considered redundant or extraneous (e.g items 3, 4 and 5 of Table 4.8) were removed. An application of the revised version (version 1.2) of the checklist to present a characterization of its elements is reported in this section using 30 of the studies assessed in Section 4.2.3. Apart from being a sample application of the checklist, this exercise also adds credence to the reproducibility assessment of the studies by investigating whether information useful to reproduction of the studies as suggested in the checklist was made available in the studies. A 'Y' was recorded if information was found, an 'N' if no (useful) information was found and an 'X' if the element was not relevant in the context of a particular study.

A compressed result of the application is presented in Table 4.9 with more details shown in Table 4.10. The summary from Table 4.9 is further presented in a grouped bar chart (Figure 4.2) to visually project the distribution and any correlation between (or across) the different entries of the aspects.

The results according to each item of the checklist are analysed below:

Dataset: The summary presented in Table 4.9 (with more details in Table 4.10 and Figure 4.2) shows that 26 (87%) of the studies provided information on the original location of the raw dataset they used but only three (10%) shared a local copy of the dataset while none of the studies made the subset, restructured or cleaned dataset they eventually used for their studies available.

Preprocessing: The details regarding the conduct of the preprocessing activities which included stopwords removal, stemming, feature representation etc. was found in 17 (57%) of the studies while 21 (90%) of the studies superficially discussed their feature representation approach.

Dimensionality reduction: Though, dimensionality reduction is a key TM process due to the generation of large but sparse feature vector during preprocessing but the typical benchmark datasets size in SRs (particularly, the ones used in the studies reviewed) are relatively small compared to what is obtainable in image classification

benchmark datasets. This may be why 25 (83%) of the studies did not report conducting any activity to reduce the dimension of their feature vector. But, five (17%) did reduce the dimension of their vector but only three (10%) gave an account of how they went about it. None of the studies made a copy of their final feature vector available for independent use while only one ((A. M. Cohen et al., 2006)) provided intermediate preprocessing output that can be used for future reference.

Data partition: None of the studies provided any information on the portions of data used for either training or testing beyond basic ratio information.

Model training: All the studies provided some details about the training of their models. However, of the 17 (57%) that proposed some new techniques, none of them provided access to their technique's code, four (13%) provided an algorithm of their techniques, only one (3%) made executable file available while 16 (53%) provided only a textual description of their techniques.

Model assessment: All the studies were able to describe how their models were assessed.

Table 4.9: Summary of the Assessment of 33 studies based on the checklist (version 1.2)

Item No.	Elements	Yes	No	N/A
1	Original location of the raw dataset	26	4	0
	Provided link to local copy of:			
2	a. Raw dataset	3	27	0
	b. Target dataset	0	0	0
3	Pre-processing details	17	13	0
4	Feature representation technique	21	9	0
5	Feature selection technique	8	19	3
6	Dimensionality reduction technique	3	2	25
7	Final feature vector – download link	0	30	0
8	Training algorithm	30	0	0
	Custom algorithm			
	a. Text	16	1	13
9	b. Code	0	16	14
	c. Algorithm	4	12	14
	d. Executable file	1	15	14
10	Model assessment method	30	0	0
11	Detailed model assessment result	30	0	0
12	Randomization seed values	0	28	2
	Training/test data partition available or indices provided			
13	a. Link to data partitions provided	0	30	0
	b. (link to) data indices provided	0	30	0
	c. Seed value provided	0	30	0
	Software information			
14	a. Name provided	23	6	1
	b. Version details	0	29	1

Table 4.10: Checklist (version 1.2) application on 30 studies for essential reproduction information

Item No.	Elements	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Original location of the raw dataset	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N
	Provided link to local copy of:															
2	a. Raw dataset	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N
	b. Target dataset	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
3	Pre-processing details	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	N	N	N	Y	N
4	Feature representation technique	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	N	N	Y	N
5	Feature selection technique	Y	N	X	X	X	X	X	Y	Y	X	X	X	N	X	X
6	Dimensionality reduction technique	X	N	X	X	X	X	Y	X	X	X	X	X	X	X	X
7	Final feature vector — download link	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
8	Training algorithm	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	Custom algorithm															
	a. Text	Y	Y	X	Y	X	X	X	X	Y	X	Y	X	X	Y	Y
9	b. Code	N	N	X	N	X	X	X	X	X	X	N	X	X	N	N
	c. Algorithm	N	N	X	Y	X	X	X	X	X	X	N	X	X	N	N
	d. Executable file	N	N	X	N	X	X	X	X	X	X	N	X	X	N	N
10	Model assessment method	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
11	Detailed model assessment result	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
12	Randomization seed values	N	N	N	N	N	N	N	N	N	N	N	N	N	X	X
	Training/test data partition available or indices provided															
13	a. Link to data partitions provided	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	b. (link to) data indices provided	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	c. Seed value provided	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	Software information															
14	a. Name provided	N	Y	N	N	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y
	b. Version details	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

Table 4.10: Checklist (version 1.2) application on 30 studies (continued)

Item No.	Elements	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	Original location of the raw dataset	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
	Provided link to local copy of:															
2	a. Raw dataset	N	N	N	N	N	N	N	N	N	N	Y	Y	N	N	N
	b. Target dataset	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
3	Pre-processing details	Y	Y	Y	N	N	N	Y	N	Y	N	Y	N	N	N	Y
4	Feature representation technique	Y	N	Y	Y	Y	Y	N	N	Y	Y	N	Y	Y	Y	Y
5	Feature selection technique	Y	Y	Y	Y	X	X	X	N	X	X	Y	X	X	X	X
6	Dimensionality reduction technique	X	X	Y	X	X	X	X	N	Y	X	X	X	X	X	X
7	Final feature vector — download link	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
8	Training algorithm	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	Custom algorithm															
	a. Text	X	Y	N	X	Y	X	Y	Y	Y	X	Y	Y	X	Y	Y
9	b. Code	X	N	N	X	N	X	N	N	N	X	N	N	X	N	N
	c. Algorithm	X	N	N	X	N	X	N	N	N	X	Y	Y	X	N	Y
	d. Executable file	X	N	N	X	N	X	N	N	N	X	N	N	X	Y	N
10	Model assessment method	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
11	Detailed model assessment result	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
12	Randomization seed values	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	Training/test data partition available or indices provided															
13	a. Link to data partitions provided	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	b. (link to) data indices provided	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N
	c. Seed value provided	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N
	Software information															
14	a. Name provided	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	X	Y	Y	Y
	b. Version details	N	N	N	N	N	N	N	N	N	N	N	X	N	N	N

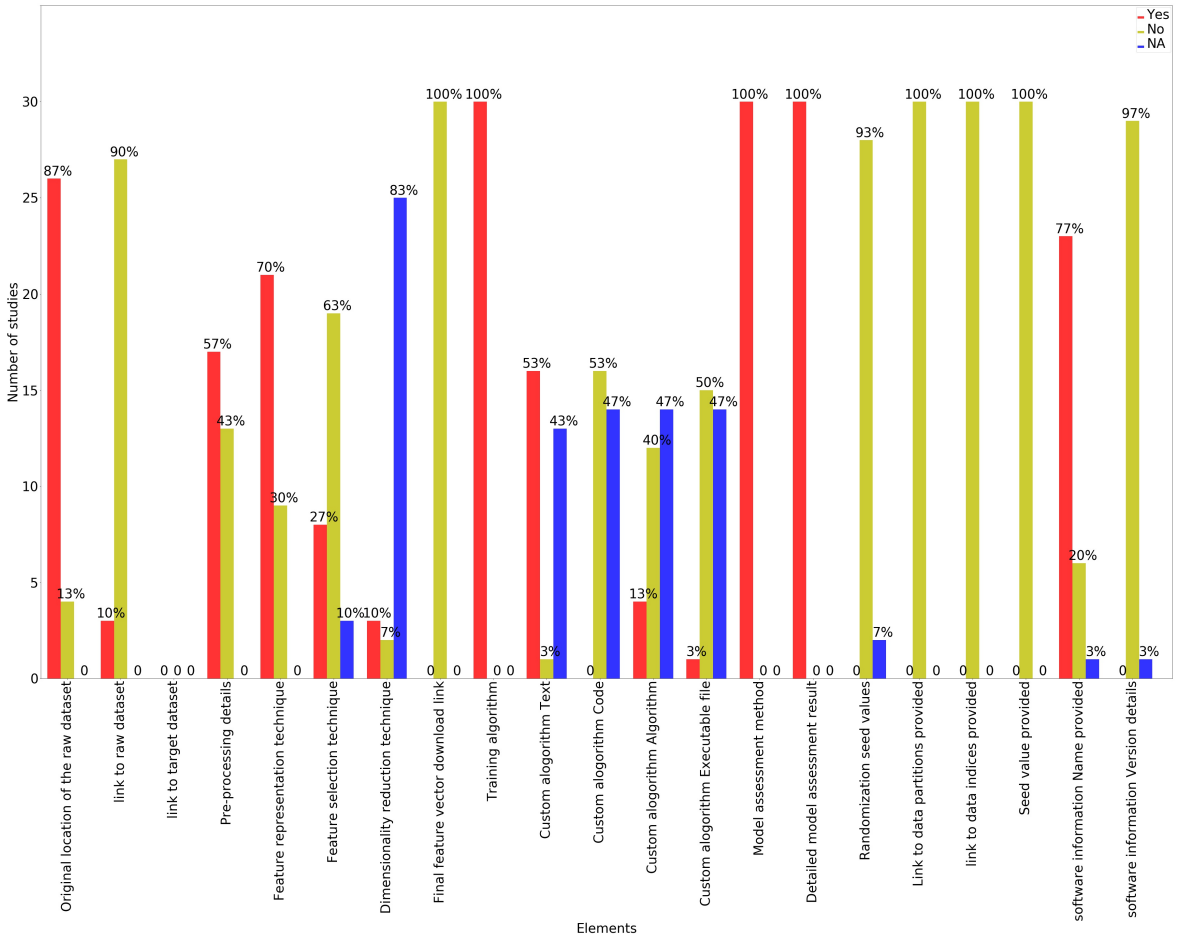

Figure 4.2: Distribution of studies containing information to support reproducibility

Randomization control: 28 (93%) of the studies performed operations that involved some randomization in the algorithm execution. However, none of them provided any information on how this was handled.

Software information: The studies generally ($\sim$ 75%) provided the main software used in their studies. Where they all failed (100%) was in providing the particular details of associated modules and packages as well as their respective version information.

4.5.3 Conclusions from the checklist application

The application of the checklist (version 1.2) on the 30 studies as summarized in Table 4.9 showed that the major points of reproducibility failure are related to:

i) Access to target dataset: the copy of the dataset(s) used (Table 4.9, item 2) by studies. All the entries has zero value, consequently, no bar in Figure 4.2. The exact copy of dataset used in a study is particularly important as dataset host site or location may become inaccessible at any time. A good example of

this situation is the case of invalid address discovered for the EPC dataset twice within the course of this research as reported in Section 4.2.1.

ii) Custom method: the new methods proposed in the studies (Table 4.9, item 9). Providing access to the implementation or executable files of the proposed methods will go a long way to ensure that ambiguities and misinterpretations are eliminated during the reproduction process as against mere text description.

iii) Randomization control: this refers to the seed values (or any other techniques used) to control randomization involved in the studies (Table 4.9, item 12). Even if every other piece of information required is provided, the presence of similar seed values (where necessary) as used in the original study is the only way to ensure the same process is repeated exactly as before.

iv) Partitioning information: the data partitions (Table 4.9, item 13) used at different stages of the study. This is essential as found for example in image recognition datasets like the Canadian Institute For Advanced Research (CIFAR) or Modified National Institute of Standards and Technology (MNIST) image datasets where the testing and training sets are provided for uniformity and comparability across experiments. Training a model with different sets of data has the potential to alter the outcome of what the model learned.

v) The names and version numbers of the different modules and packages contained in the software environment used for the studies (Table 4.9, item 14b) of the table.

The artefacts in this list are critical to reproduction. It could be possible to reproduce some of the other reproducibility factors not in the list above if not provided supposing the items in this list are provided.

The checklist application revealed that less attention was paid to the provision of datasets for replication use. Apart from access to the raw dataset, providing access to the different partitions used for training, evaluating or testing purposes had not been given proper attention. As an alternative, with sufficient information and access to ordered dataset, seed value information and algorithms used for the partition will be sufficient but it can be seen in Figure 4.2 that the assessed studies failed to provide these essential information.

According to Table 4.9, researchers usually provide the name of the dataset or its host. It should be realized that providing the name of a popular dataset or that of its provider may sometimes be insufficient to have studies reproduced. Beyond the raw dataset, there may be need for extraction of part and even cleaning of the retrieved subset. Independent researchers should be able to get hold of the exact (ordered) replica, of the dataset used in studies else reproduction may be impossible. Therefore, rather than give data or host name, it is more appropriate to provide

access to the subset of the data that was used in particular experiments since most of the available dataset like the TREC are usually large and hardly used completely in a single experiment. Otherwise, a link to the raw dataset, access to the code used for extracting the portion used and details of the fields used will suffice.

Given the constant maintenance and updates of software packages, it is important to provided specific details of the software environment used during the course of a study (Sandve et al., 2013). A notable example is the deprecation of the module used for CV in Python's 'sklearn' (version 0.17), the *cross_validation* module was discontinued for the *model_selection* module in version 0.18 upwards to perform similar function but with a different interface. It was a similar situation for the 'auto' option for the *class_weight* parameter (to cater for class imbalance) in most *sklearn's* classification modules which is now deprecated for the 'balanced' option. On the same dataset both *class_weight* options will produce different results. Other examples include the current changes in the various interfaces of Keras 2.0 compared to previous versions.

A further revision and validation of the checklist (version 1.3) on new studies is presented in Section 8.3.1.

4.6 Summary

The study reported in this chapter has evaluated the potential for being reproducible in 33 studies using a systematic qualitative assessment approach. A set of elements of the study process, attributes of the elements, values to provide a measure of the attributes and tags to summarise or judge the overall observation were defined for the reproducibility assessment. In addition, the checklist was applied on the 30 studies to visually explore how the listed elements are characterized.

Considered as a whole, the results of this study indicated that the assessed studies cannot be fully reproduced. However, if the different steps of the studies are considered separately, each of them exhibited different levels of reproducibility. The third party frameworks used were the easiest to identify and access followed by the datasets used which can be located with some difficulties similar to the case of reproducing the training and testing methods used in studies. Explicit information concerning datasets, study parameters (particularly randomization control) and software environment are lacking in most studies and consequently hinder their reproducibility. It was also found that when researchers propose new methods, they only explain it in the study and at best provide some form of algorithms about it. Code implementations and/or executable files are usually not made available for the community's future use. The field thrives on the availability of public datasets; therefore, researchers should do more by making their knowledge more accessible for easier development and advancement of the body of knowledge.

The results of this study have provided a useful insight into the information required to be captured in TM study reports to enhance their reproducibility. This has led to the creation of a preliminary version of a checklist in this study. Details of a novel transparent tool developed to screen citation for SRs purpose are provided in a later chapter (Chapter 7). Parts of its design were informed by this study, the review presented in Chapter 3, investigation of model complexity and role of feature representations in Chapter 5 and investigation into feature enrichment presented in Chapter 6.

CHAPTER 5

Reporting model complexity in CS studies

Lack of technical information (relating to study processes and models), provided in studies on automatic screening of citations using TM techniques was identified in Chapter 3. This raised a concern about the possible effect of this lack of information, particularly in connection to the fundamentals of scientific research requirements vis-á-vis study reproducibility which is a general requirement and model complexity which is particular to computational studies. A study assessing the effect of the lack of technical information relating to study processes was reported in Chapter 4. 33 studies were assessed for reproducibility and the information provided was found to be insufficient to reproduce any of the studies. The information needed to support reproducibility was subsequently identified and combined into a checklist. This chapter investigates the effect of the lack of technical information relating to models. To this end, the complexity of the models was investigated to substantiate the need to include information related to model complexity in the study reports for more transparency. The complexity of ML models as used in this study generally refers to the consideration of the number of parameters a model uses in its decision making and it's hypothesis function relative to the size of its training data size (Rissanen, 1983; Myung, 2000). This is usually indicative of the models' quality and how well it generalize over the datasets.

This chapter develops and assesses the complexity of hypothetical SVM models with performances representative of those found in the literature. The complexity was measured based on the number of SVs each model used for decision making compared to their training data sizes. The results of this study show that even if a study exhibited a high classification performance, it might still have a high complexity which is in turn a concern about its generalizability over the data as the classification performance might have, for instance, been due to overfitting. The study therefore recommends that studies provide complexity related information about their models alongside classification performance results. The findings of this study have been reported in (Olorisade et al., 2017a).

5.1 Introduction

The study reported in this chapter addressed the need to provide information related to the complexity of the models. The study developed SVM models and assessed the number of SVs used by the models in decision making alongside the classification performances. In a Support Vector Machine (SVM) model, the number of SVs is the determinant of its complexity (See Section 2.1.1 and Figure 2.1 for more details on SVs and SVM). 19 datasets – four from software engineering and 15 from the healthcare research – were used. The four SE datasets are from reviews by Kitchenham[1], Hall (T. Hall, Beecham, Bowes, Gray, & Counsell, 2012)[2], Wahono (Wahono, 2015) and Radjenovic (Radjenovi, Heriko, Torkar, & ivkovi, 2013). The 15 medical reviews datasets are from the DERP dataset (A. M. Cohen et al., 2006; A. M. Cohen & Yen, 2014). The effect of feature representation techniques was also investigated in the study by using two approaches to represent the features. The binary and Word2vec feature representation techniques were used. The models were built using the cross validation approach, each model was assessed using the mean of the recall, precision and the number of SVs used.
Since it was not possible to fully reproduce any of the existing studies as reported in Chapter 4 (Sections 4.3.1 and 4.3.3), SVM models with performance representative of what is obtainable in the existing studies were built for the purpose of this study. The SVM algorithm has been chosen for this study because it was one of the most widely used, used in 31% of studies on the automation of CS as reported in Section 3.4.2.

The rest of this chapter is structured as follows: A brief background on model selection is presented in Section 5.2 while the methodology of the study is focussed on in Section 5.4. The outcomes of the study are highlighted in Section 5.5, followed by the study limitations in Section 5.7 and discussions in Section 5.8. The chapter is concluded with a summary in Section 5.9.

5.2 Model selection and complexity

The relevance of model selection in classifiers was mentioned in Section 2.2.5. The option of selecting the best model in ML is not usually a straightforward one. The rule of thumb is to select a model with the least generalization error (see Sections 2.2.5 and 2.4 for relevant discussions on model selection practices). A good model is one with low generalization error and low tendency to overfit (Nannen, 2010; Rissanen, 1983). Given two possible representations or models of data, the Occam's razor prin-

[1]This dataset is provided by Prof. B. Kitchenham

[2]The Hall, Wahono and Radjenovic datasets were recreated and used in (Z. Yu et al., 2016) and made available by its author.

ciple dictates that, all other things being equal, the simpler or less complex of the two should be preferred (Domingos, 1999b, 2012). Consequently, the understanding of this principle has generated a few controversies based on different interpretations and drawing of unsupported conclusions between simplicity and accuracy (Domingos, 2012). Simplicity in this context refers to the representation generated from a less complex hypothesis (Blumer, Ehrenfeucht, Haussler, & Warmuth, 1987; Domingos, 1999b), which may be easier to understand, and/or explained.

There are a number of ways to determine the complexity of a model, such as Minimum Description Length (MDL) (Grünwald, 2000; Hansen & Yu, 2001) and Kolmogorov complexity (Chaitin, 1969; Kolmogorov, 1965; Solomonoff, 1964). The MDL seeks a model that yields a suitable balance between model accuracy and complexity given the sample size and data complexity (A. Barron, Rissanen, & Yu, 1998). Kolmogorov complexity is the length of the shortest program a finite string can be computed from (Grünwald, 2000). Originally used in information theory, it is lately becoming more popular in computational studies. In essence, applied to computational models, it implies the simplest hypothesis a data can be represented or approximated from. In the case of models with multiple components, the often adopted measure of complexity is the measure of structural complexity, which is given by the number of components of the model. This approach is valid in particular, when each component of the model can be expected to have the same level of complexity as any other component of the model.

5.3 Complexity in SVM classification models

The number of SVs is indicative of the complexity of the SVM classification models (Cristianini & Shawe-Taylor, 2000; Cortes & Vapnik, 1995). The statistical theory of the SVM is based on the assumption that the algorithm uses as few SVs as possible to make its decision (see Figure 2.1). A significant advantage of the SVM is that since it requires only a small number of SVs, it is robust to small sample sizes or situations where the number of features is more than the number of samples because it needs only a few of the samples as SVs (Baharudin et al., 2010; Ikonomakis et al., 2005). In order to benefit as much as possible from this aspect of the SVM classification, it is important to assess a range of options for the SVM classifiers, including different kernels and complexity penalty values ('C') - parameter selection (Cherkassky & Ma, 2004b). The generalisation error of a SVM is proportional to the ratio of the dimension of the data and the total number of vectors and also grows with the dimensionality of the data (Niyogi & Girosi, 1999; A. R. Barron, 1993; MIT, 2007; Abu-Mostafa, 2012; Bartlett & Shawe-Taylor, 1999; Steinwart, 2003).

In SVM, the higher the number of SVs, the more complex the model is and the higher the possibility of classification error and overfitting. According to A. Barron

et al. (1998), learning in models is a function of the hypothesis, representation and optimization. There is hardly any optimization done by a SVM model, when (almost) all the dataset acts as SVs. Such models are almost equivalent of a nearest neighbour classifier using all available training data. Consequently, the statistical validity of SVM models, where a large fraction of the training data constitute SVs, is comparable to the statistical validity of nearest neighbour classifiers based on the full training data. As new/more data, advanced knowledge and improved algorithms become available, complexity details is key to determining if new models are actually better and smarter or just fitting to the data noise like (most of) the ones shown in this study.

5.4 Complexity assessment

This study was conducted following the basic text mining steps as highlighted in Section 2.2 (depicted in Figure 2.4). The models were developed using the SVM method. Each model was developed using two feature representation types - binary and the Word2vec representations, based on the four reviews datasets from SE (Kitchenham, Hall, Wahono and Radjenovic) and 15 datasets from medical reviews (A. M. Cohen et al., 2006; A. M. Cohen & Yen, 2014). The study process consisted of steps as described in Section 2.2. The core of this study was conducted using methods from the 'sklearn', the Natural Language Toolkit - NLTK and the gensim packages from Python with some custom codes particularly for sentence parsing for the Word2vec representation.

The goal of this study was to investigate complexity concerns about TM models for automatic screening of citations in SRs and establish the need to report relevant information about them alongside other performance measures. The investigation was conducted by developing SVM models with similar performance to what is obtainable in the literature and measure their complexities.

5.4.1 Data retrieval

Three of the software engineering datasets (Hall, Wahono and Radjenovic) were from previous SRs recreated for the purpose of automatic CS study by (Z. Yu et al., 2016). The Kitchenham dataset was also from a previous SR, labelled and made available by one of the authors, Prof. Barbara Kitchenham. The 15 healthcare datasets as mentioned in Section 5.1 are part of the DERP datasets and made open as part of the TREC datasets. However, in contrast to the retrieval approach for the same dataset described in Section 4.2.1.1, an alternative source was explored in this study.

The articles' PMID provided by Cohen et al. in a supplementary file[3] to (A. M.

[3]http://skynet.ohsu.edu/~cohenaa/systematic-drug-class-review-data.html

Cohen et al., 2006) guided the identification and retrieval of relevant papers from the "PubMed" database. It was noticed that one of the PMIDs is no longer valid[4]. The studies and the number of documents in each review are presented in Table 5.1. The normalized distribution of the positive to negative samples is presented in Fig. 5.1.

Table 5.1: Corpus retrieved for each review

Review	Corpus size	Negative class	Positive class
Kitchenham	1704	1659	45
Hall	8911	8805	106
Wahono	7002	6940	62
Radjenovic	6000	5962	48
ACEinhibitor	2544	2503	41
ADHD	851	831	20
Antihistamines	310	294	16
AtypicalAntipsychotics	1120	974	146
BetaBlockers	2072	2030	42
CalciumChannelBlockers	1218	1118	100
Estrogens	368	288	80
NSAIDs	393	352	41
Opioids	1915	1900	15
OralHypoglycemics	503	367	136
ProtonPumpInhibitors	1333	1282	51
SkeletalMuscleRelaxants	1643	1634	9
Statins	3465	3380	85
Triptans	671	647	24
UrinaryIncontinence	327	287	40

5.4.2 Text preprocessing

Prior to the commencement of the preprocessing step, the datasets were initially shuffled. Preprocessing in this study involved the process of splitting the text into

[4]This is a possibility hinted by Cohen et al. on the webpage. "12168612" updated to "11757504"

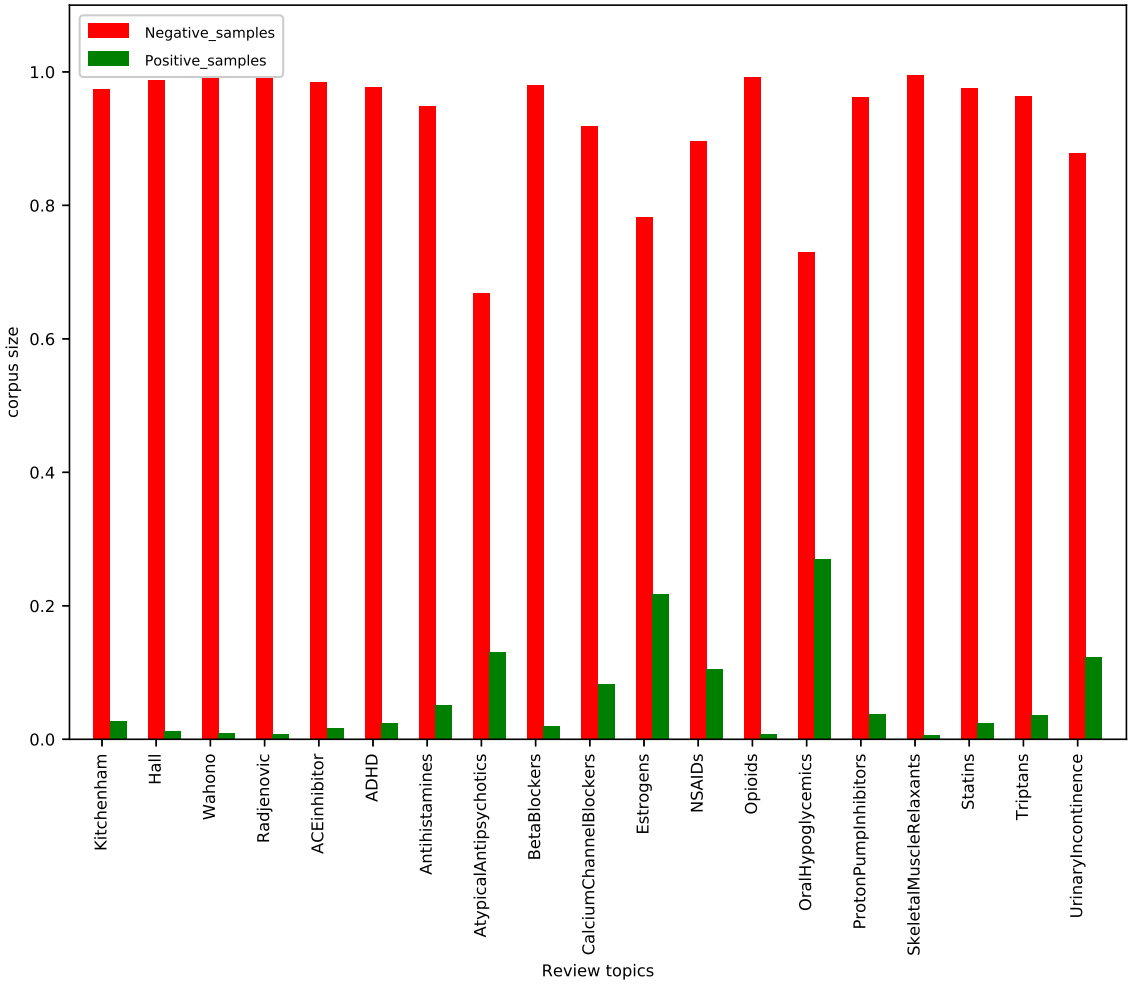

Figure 5.1: Normalized distribution of relevant-irrelevant candidate articles

individual words (tokenization), removing unwanted words (stopwords) and characters (numbers, special characters etc.), creating a "Dictionary" from the corpora by merging all the corpus' texts (for each dataset) and numerically encoding the text data (see Section 2.2.2). The healthcare data consisted of the articles' titles, abstracts and the Medical Subject Heading (MeSH) as was used in (A. M. Cohen et al., 2006) while the SE data consisted of only the titles and abstracts. The English stopwords were removed and only the features that appeared in a minimum of two documents and in a maximum of 80% of the corpus were retained. Following this step was the initial splitting of the datasets for the purpose of selecting optimal parameters for training the final model. 70% of each dataset was used for selecting optimal parameters for the models where the data size was above 1000 and 80% otherwise. The seed values used across this study were chosen randomly for experimental and reproduction purposes (in line with recommendations from Section 4.5), since none of the previous studies in the field reported any. The number of features resulting from this operation for each of the studies is shown in Table 5.2.

Table 5.2: Top Selected Features

Review	Feature size	Selected top features
Kitchenham	5436	267
Hall	11781	589
Wahono	10895	545
Radjenovic	9860	493
ACEinhibitor	5754	210
ADHD	3591	80
Antihistamines	2105	29
AtypicalAntipsychotics	4131	381
BetaBlockers	5567	194
CalciumChannelBlockers	4111	329
Estrogens	2489	233
NSAIDs	2409	242
Opioids	5512	55
OralHypoglycemics	2759	234
ProtonPumpInhibitors	3942	206
SkeletalMuscleRelaxants	5835	11
Statins	7240	467
Triptans	3035	121
UrinaryIncontinence	2315	215

5.4.3 Feature representation

The features were represented in both binary and Word2vec forms (see Section 2.2.3 for definitions of these representations). The packages used for the preprocessing of both features as described in Section 5.4.2 are different. While the 'sklearn' packages were used for the binary representation, the NLTK package (Bird, 2006) with custom codes was used to preprocess the Word2vec representation. After removing the stopwords and tokenizing the texts, a word to vectors model was then trained with the aid of the Word2vec method in the gensim package. This model was used to transform the corpus to 'average word features'.

5.4.4 Feature selection

The dimensionality of the resulting feature vectors was reduced using the χ^2 method in the sklearn's model selection routine to select top features that were found to be significant at 0.05 α level according to the values reported in (A. M. Cohen et al., 2006) for the medical datasets. Owing to the lack of any benchmark study for the SE datasets, the same procedure was followed as closely as possible. The number of total features and of the top features retained is shown in Table 5.2.

5.4.5 Parameter selection

The top features resulting from the operations described in Sections 5.4.2 and 5.4.4 were used in a grid search cross validation (optimizing for recall) to select the best combination of SVM parameters that gave highest recall. Recall is the simplest metric that can convey to a systematic reviewer how many of the relevant articles have been correctly identified by a model. This is important in SRs (particularly in the medical domain where it may be critical to ensure all available evidence is retrieved). Thus, recall was made the primary metric to select the model parameters. The grid search CV parameter was set to two to simulate the cross validation approach that was later used for the model on the whole dataset.

5.4.6 Model training and assessment

Three types of SVM models were developed from each of the datasets:

i) binary feature with non-linear kernel

ii) Word2vec feature with linear kernel

iii) Word2vec feature with non-linear kernel

The models were trained with the best model parameters for each dataset returned from the parameter selection step described in Section 5.4.5. The models

Table 5.3: Datasets split size for cross validation

Review	Negative sample	Positive sample
Kitchenham	829	22
Hall	7044/1761	85/21
Wahono	5552/1388	5012
Radjenovic	4762/1200	38/12
ACEinhibitor	1252	21
ADHD	415	10
Antihistamines	147	8
AtypicalAntipsychotics	487	73
BetaBlockers	1015	21
CalciumChannelBlockers	559	50
Estrogens	144	40
NSAIDs	176	21
Opioids	950	8
OralHypoglycemics	184	68
ProtonPumpInhibitors	641	26
SkeletalMuscleRelaxants	817	5
Statins	1690	43
Triptans	324	12
UrinaryIncontinence	143	20

were trained and assessed using the 5X2-fold CV. The CV approach was changed to 2X5-fold for the Hall, Radjenovic and Wahono datasets because they are relatively larger. The size of train/test partitions for the negative and positive during the cross validation is shown in Table 5.3. The training and test set was shown for the Hall, Wahono and Radjenovic datasets that used the 2X5-fold CV while a single number was shown for the rest of the datasets that used used the 5X2-fold CV since the training and test are equal in each fold. The CV process was described in Section 2.4.1.

Each dataset was split with different seed values on each fold run. The recall, precision, accuracy and number of SVs were accumulated and averaged. The details required for the reproducibility of this study as guided by the finding from Chapter 4 are provided in Appendix C. The software environment information is presented in Table C.1.

In CS for SRs, full recall of all relevant studies is usually the primary target. Thus, the models with highest recall for the positive class were chosen for each review. The results for the different models given the two variants of feature representations and the 19 reviews are shown in Tables 5.4, 5.5 and 5.6. The performance metrics – recall, accuracy and precision, are the mean and standard deviation values over the ten runs. Apart from the usual recall and precision metrics, the mean and standard deviation of the number of the SVs that each of the models used in making its classification judgements is shown - this characterises the complexity as well as the statistical validity of the SVM classifiers as described in Section 5.3.

5.5 Results

The model assessment results (recall, precision and accuracy) are presented in this section alongside the models' complexity assessment indicated by the number of SVs.

Three different SVM models with similar recall performance - two built with Word2vec features (linear and non-linear kernels), and the third with binary features from non-linear kernels - were built for each review. The results of the Word2vec linear kernel, Word2vec non-linear kernel and the binary non-linear kernel models are respectively presented in Tables 5.4, 5.5 and 5.6. In addition to the metrics values, the non-linear kernel model tables (Tables 5.5 and 5.6) contain the values of three key parameters - kernel name, C and gamma.

The Word2vec linear kernel models (Table 5.4) and the non-linear kernel (rbf and sigmoid) models (Table 5.5) presented similar performances across all data sets. The linear kernels exhibited poor performance with the binary features for which the non-linear kernels showed better performance (Table 5.6). Consequently, the results of the binary features based on linear kernel models had been excluded.

5.6 Results analysis

The results of the study is further analysed in this section. In particular, the t-test method is used to test for any significance difference in the total number of SVs used by the different models from the same dataset. The result of the t-test is presented in Table 5.7. The linear kernel based (Word2vec feature) models used a significantly ($p < 0.05$) fewer SVs in nine cases than their non-linear counterpart, while the non-linear Word2vec models used fewer SVs than the linear kernel Word2vec models in four cases. There was no significant difference between the SVs used by the two models in the remaining six cases (Table 5.7).

In 13 cases, the linear kernel Word2vec models used fewer SVs than the binary feature non-linear kernel models. There was no significant difference in the SVs used

Table 5.4: Word2Vec Linear Kernel (W2V-L)

Review	Mean Performance (5x2-fold CV)			Support vectors		configuration
	precision	recall	accuracy	neg	pos	parameters[5]
Kitchenham	0.05 ± 0.01	0.91 ± 0.08	0.56 ± 0.04	1029 ± 50	10 ± 1	linear, 100
Hall	0.11 ± 0.01	0.97 ± 0.04	0.90 ± 0.01	1778 ± 185	12 ± 2	linear, 1.0
Wahono	0.07 ± 0.00	0.94 ± 0.06	0.88 ± 0.01	1609 ± 113	8 ± 1	linear, 1.0
Radjenovic	0.05 ± 0.01	0.90 ± 0.09	0.86 ± 0.03	1411 ± 184	8 ± 1	linear, 1.0
ACEinhibitor	0.07 ± 0.02	0.94 ± 0.04	0.78 ± 0.05	595 ± 88	7 ± 1	linear, 1.0
ADHD	0.08 ± 0.01	0.93 ± 0.09	0.75 ± 0.01	251 ± 43	4 ± 1	linear, 1.0
Antihistamines	0.06 ± 0.00	0.90 ± 0.12	0.22 ± 0.06	141 ± 6	4 ± 1	linear, 40.0
AtypicalAntipsychotics	0.17 ± 0.02	0.92 ± 0.05	0.40 ± 0.09	420 ± 43	25 ± 4	linear, 1000
BetaBlockers	0.05 ± 0.01	0.89 ± 0.07	0.63 ± 0.05	675 ± 68	8 ± 2	linear, 1.0
Calcium...Blockers	0.12 ± 0.01	0.92 ± 0.06	0.45 ± 0.07	477 ± 45	20 ± 2	linear, 100
Estrogens	0.32 ± 0.01	0.92 ± 0.06	0.56 ± 0.03	121 ± 8	12 ± 2	linear, 1000
NSAIDs	0.15 ± 0.02	1.00 ± 0.00	0.38 ± 0.06	157 ± 5	5 ± 1	linear, 1.0
Opioids	0.03 ± 0.00	0.81 ± 0.11	0.76 ± 0.05	482 ± 44	4 ± 1	linear, 1.0
OralHypoglycemics	0.28 ± 0.01	0.98 ± 0.02	0.30 ± 0.02	182 ± 2	33 ± 2	linear, 10000
ProtonPumpInhibitors	0.06 ± 0.01	0.91 ± 0.07	0.47 ± 0.09	542 ± 56	9 ± 1	linear, 1.0
Skeletal...Relaxants	0.01 ± 0.01	0.66 ± 0.23	0.53 ± 0.11	610 ± 83	4 ± 0	linear, 1.0
Statins	0.05 ± 0.00	0.92 ± 0.04	0.56 ± 0.06	1250 ± 83	14 ± 2	linear, 1.0
Triptans	0.06 ± 0.00	0.97 ± 0.06	0.44 ± 0.07	286 ± 16	4 ± 1	linear, 1.0
UrinaryIncontinence	0.20 ± 0.03	0.93 ± 0.07	0.53 ± 0.11	122 ± 15	6 ± 1	linear, 100

[5]Parameter — kernel, C, gamma

by the two models in one case and the binary non-linear kernel model used fewer SVs in the remaining one case. In 12 cases, the Word2vec non-linear kernel models used significantly ($p < 0.05$) fewer SVs than the binary features non-linear kernel models (Table 5.7). In two cases, the binary non-linear kernel models used significantly ($p < 0.05$) fewer SVs than the Word2vec non-linear kernel models and there was no significant difference between the two in five cases (Table 5.7).

The number of SVs in general was large in the resulting models from the data in this study. Practical experience has shown that the appropriate ratios for good generalisation are typically 2% – 5% and less than 2% for large volumes of data. Natural Language Processing (NLP) data is usually high dimensional and the number of data points relative to the dimensionality of the data is relatively small. Given the previously noted reasons, i.e. that the generalisation error of SVMs grows with the number of SVs and the dimensionality of the data, it is very important for NLP applications of SVMs to keep the number of SVs low. The fact that many more SVs were found in the study's SVM classifiers may mean that in this case the SVM optimization was

Table 5.5: Word2Vec Non-linear Kernel (W2V-NL)

Review	Mean Performance (5x2-fold CV)			Support vectors		configuration
	precision	recall	accuracy	neg	pos	parameters[6]
Kitchenham	0.04 ± 0.00	0.97 ± 0.05	0.31 ± 0.12	1314 ± 16	10 ± 0	rbf, 1000
Hall	0.11 ± 0.01	0.97 ± 0.04	0.90 ± 0.01	1778 ± 185	12 ± 2	sigmoid, 1000
Wahono	0.07 ± 0.00	0.94 ± 0.06	0.88 ± 0.01	1609 ± 113	8 ± 1	sigmoid, 1000
Radjenovic	0.03 ± 0.01	0.97 ± 0.05	0.76 ± 0.03	2558 ± 136	9 ± 1	sigmoid, 100
ACEinhibitor	0.08 ± 0.02	0.92 ± 0.06	0.82 ± 0.05	488 ± 96	7 ± 1	rbf, 1000, .001
ADHD	0.06 ± 0.01	0.99 ± 0.03	0.60 ± 0.005	386 ± 20	5 ± 1	rbf, 1000, .001
Antihistamines	0.05 ± 0.00	0.90 ± 0.12	0.19 ± 0.06	142 ± 6	4 ± 1	sigmoid, 1000
AtypicalAntipsychotics	0.15 ± 0.01	0.97 ± 0.04	0.26 ± 0.09	476 ± 5	16 ± 3	sigmoid, 10000, .001
BetaBlockers	0.05 ± 0.01	0.89 ± 0.07	0.63 ± 0.05	675 ± 68	8 ± 2	sigmoid, 1000, .001
Calcium...Blockers	0.11 ± 0.01	0.94 ± 0.06	0.35 ± 0.10	517 ± 37	20 ± 2	sigmoid, 10000
Estrogens	0.26 ± 0.02	0.97 ± 0.04	0.39 ± 0.08	138 ± 4	12 ± 1	sigmoid, 10000
NSAIDs	0.18 ± 0.02	1.00 ± 0.01	0.53 ± 0.06	139 ± 10	5 ± 1	sigmoid, 1000
Opioids	0.01 ± 0.00	1.00 ± 0.00	0.45 ± 0.05	890 ± 25	4 ± 1	sigmoid, 10
OralHypoglycemics	0.27 ± 0.03	1.00 ± 0.00	0.27 ± 0.00	183 ± 1	40 ± 7	sigmoid, 1000
ProtonPumpInhibitors	0.04 ± 0.00	0.97 ± 0.04	0.15 ± 0.12	637 ± 7	9 ± 1	sigmoid, 100, .001
Skeletal...Relaxants	0.01 ± 0.00	0.96 ± 0.12	0.29 ± 0.13	763 ± 50	4 ± 1	rbf, 100, .001
Statins	0.03 ± 0.00	0.99 ± 0.01	0.26 ± 0.07	1647 ± 27	14 ± 1	sigmoid, 100, .001
Triptans	0.06 ± 0.01	0.98 ± 0.05	0.40 ± 0.07	292 ± 15	4 ± 1	sigmoid, 100
UrinaryIncontinence	0.20 ± 0.04	0.93 ± 0.07	0.52 ± 0.12	123 ± 15	6 ± 1	rbf, 10000

[6]Parameter — kernel, C

complicated and slow, which eventually lead to an early stop of the optimizers before achieving any significant optimization. Consequently, the statistical validity of these results was likely to be relatively limited, or in other words the likely error bounds were large and the likelihood of wrong classifications was also relatively large. The models from three of the SE datasets (Hall, Wahono and Radjenovic) might have however learnt better and show lower complexity since they used only about 30% of their training data as SVs. This might be because their data size is larger than the rest, so the model had enough examples to learn from and make a more robust decision relying on fewer number (ratio) of data samples.

Table 5.6: Binary Non-linear Kernel (B-NL)

Review	Mean Performance (5x2-fold CV)			Support vectors		configuration
	precision	recall	accuracy	neg	pos	parameters[7]
Kitchenham	0.06 ± 0.01	0.90 ± 0.12	0.61 ± 0.01	1326 ± 3	26 ± 2	rbf, 1.0
Hall	0.27 ± 0.03	0.94 ± 0.05	0.97 ± 0.00	2813 ± 173	41 ± 2	sigmoid, 1.0
Wahono	0.18 ± 0.02	0.86 ± 0.06	0.96 ± 0.00	2102 ± 84	36 ± 2	sigmoid, 1.0
Radjenovic	0.14 ± 0.03	0.76 ± 0.10	0.96 ± 0.01	1958 ± 79	28 ± 1	sigmoid, 1.0
ACEinhibitor	0.12 ± 0.02	0.82 ± 0.08	0.90 ± 0.02	964 ± 104	16 ± 1	sigmoid, 1.0, .001
ADHD	0.12 ± 0.01	0.95 ± 0.05	0.83 ± 0.02	390 ± 31	9 ± 1	rbf, 1.0, .001
Antihistamines	0.06 ± 0.03	0.53 ± 0.40	0.50 ± 0.37	146 ± 3	7 ± 1	rbf, 10, .001
AtypicalAntipsychotics	0.23 ± 0.03	0.86 ± 0.07	0.59 ± 0.09	477 ± 14	47 ± 3	rbf, 1.0, auto
BetaBlockers	0.06 ± 0.02	0.82 ± 0.16	0.67 ± 0.24	986 ± 63	14 ± 1	sigmoid, 1.0, .001
Calcium...Blockers	0.22 ± 0.04	0.77 ± 0.06	0.74 ± 0.06	499 ± 25	34 ± 2	rbf, 1.0, auto
Estrogens	0.39 ± 0.03	0.93 ± 0.07	0.66 ± 0.04	136 ± 3	27 ± 1	rbf, 1.0, auto
NSAIDs	0.30 ± 0.04	0.93 ± 0.03	0.76 ± 0.04	173 ± 3	14 ± 1	rbf, 10, .0001
Opioids	0.10 ± 0.09	0.74 ± 0.22	0.73 ± 0.37	947 ± 9	7 ± 1	sigmoid, 1.0, .001
OralHypoglycemics	0.28 ± 0.01	0.97 ± 0.05	0.33 ± 0.04	183 ± 1	43 ± 3	sigmoid, 1.0, .001
ProtonPumpInhibitors	0.10 ± 0.01	0.82 ± 0.12	0.71 ± 0.07	602 ± 48	17 ± 2	rbf, 1.0, .001
Skeletal...Relaxants	0.00 ± 0.00	0.01 ± 0.00	1.00 ± 0.00	817 ± 0	5 ± 1	rbf, 1.0, .001
Statins	0.08 ± 0.02	0.82 ± 0.09	0.75 ± 0.07	1439 ± 147	31 ± 2	sigmoid, 1.0, .001
Triptans	0.12 ± 0.02	0.82 ± 0.12	0.76 ± 0.09	309 ± 31	10 ± 1	rbf, 1.0, .001
UrinaryIncontinence	0.37 ± 0.06	0.80 ± 0.08	0.80 ± 0.05	135 ± 10	15 ± 2	sigmoid, 1.0, auto

[7]Parameter — kernel, C, gamma

5.7 Threats to study validity

The study is mainly limited by threats affecting its external validity caused by the specific datasets used, the values of the parameters and data partitions. The performances of the SVMs used in this study are not necessarily generalizable. The majority of the sample sizes are quite small with considerably imbalanced classes. Though, this characteristics is representative of the real life systematic reviews datasets and attempts were made to reduce the effect of a one-off result through cross validation and averaging the performance results.

During the grid search for the model with the best recall, the results were limited to the bound of values set for each parameter. It was impractical for us to exhaust

Table 5.7: Paired t-test result for difference in number of SVs

Study No.	W2V-L vs W2V-NL	W2V-L vs B-NL	W2V-NL vs B-NL
Kitchenham	0.0000*,-	0.0042*,-	0.0051*,-
Hall	0.8905	0.0042*,-	0.0051*,-
Wahono	0.9985	0.0042*,-	0.0051*,-
Radjenovic	0.0000*,-	0.0042*,-	0.0.0051*,+
ACEinhibitors	0.0248*,+	0.0000*,-	0.0000*,-
ADHD	0.0000*,+	0.0000*,-	0.489
Antihistamines	0.0000*,-	0.0019*,-	0.0026*,-
AtypicalAntipsychotics	0.0319*,+	0.0005*,+	0.0048*-
BetaBlockers	1.00	0.0000*,-	0.0000*,-
Calcium...Blockers	0.0615	0.0583	0.793
Estrogens	0.0004*,-	0.0000*,-	0.0000*,-
NSAIDS	0.0002*,+	0.0000*,-	0.0000*,-
Opioids	0.0000*,-	0.0000*,-	0.0000*,-
OralHypoglycemics	0.0043*,-	0.0000*,-	0.375
ProtonPumpInhibitors	0.0007*,-	0.0136*,-	0.132
Skeletal...Relaxants	0.0003*,-	0.0000*,-	0.0098*,-
Statins	0.0000*,-	0.0025*,-	0.0036*,+
Triptans	0.4704	0.0333*-	0.0745
UrinaryIncontinence	0.9361	0.0042*-	0.0051*,-

* indicates cases where there is significant difference based on p-values (<0.05);
+ indicates the first model used more number of SVs;
- indicates the first model used fewer number of SVs.

all viable values, especially for 'C' and gamma. The values were however carefully chosen around the set of values where the models have shown to exhibit better performance and control overfitting. Similarly, it was impractical to explore setting the seed to every possible values beyond the ones reported, therefore the outcome of all randomization possibilities could not be explored and their effect on data partitions and the results. Other seed values could have had effect on the results observed for recall and precision, the effects may not have been significant and there was no indication the number of SVs used by each model could have been significantly reduced due to this.

5.8 Discussion

The results of this study and its analysis show that the models with binary features and non-linear kernel, show a higher number of SVs than the Word2vec based models that were built using the linear and non-linear kernels - Figures 5.2 and 5.3 show respectively, bar plots of normalized ratios of the number of negative and positive SVs used by each model in each review.

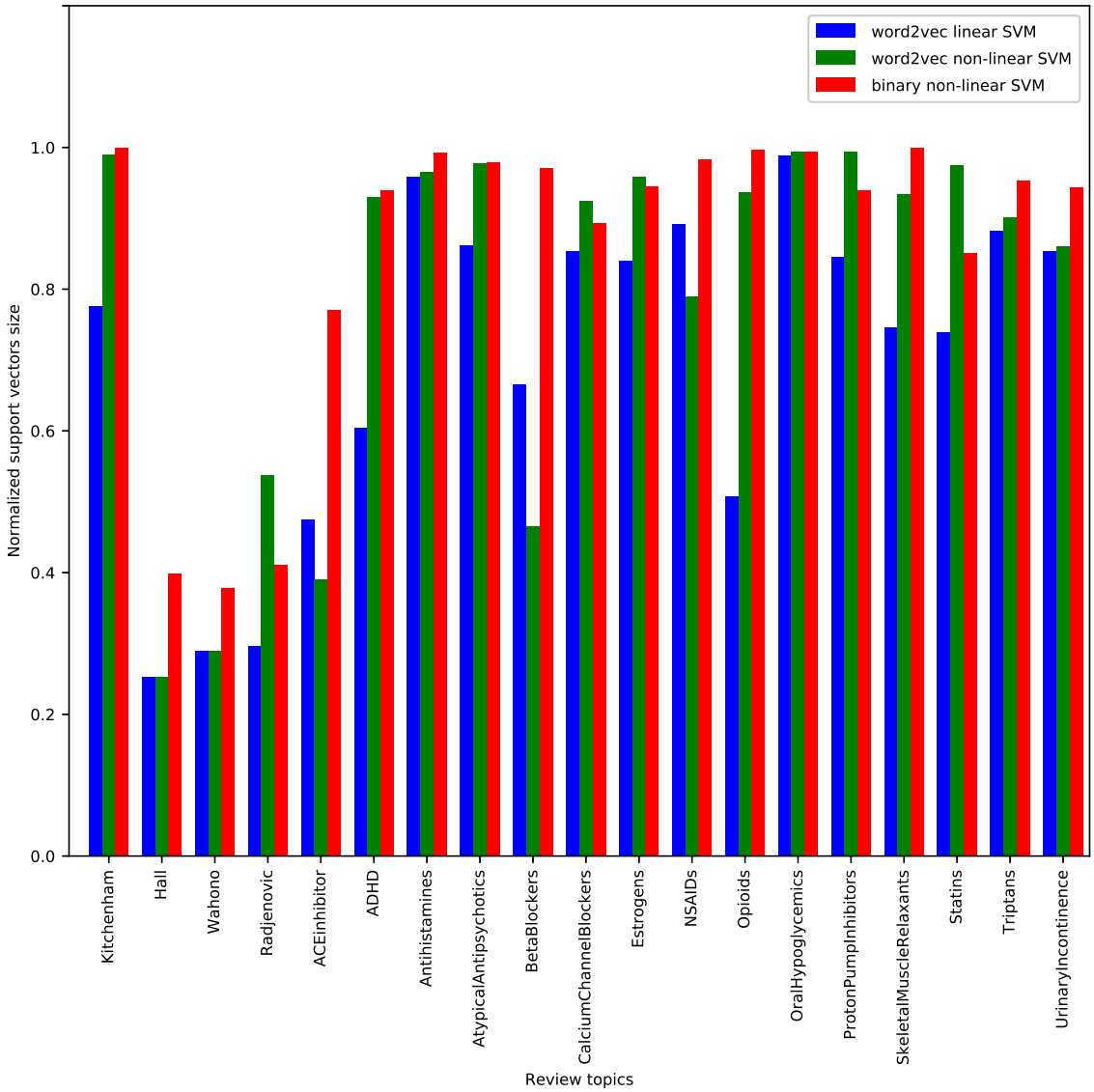

Figure 5.2: Normalized size ratio of negative SVs

This difference in the number of SVs is particularly noticeable in cases where the sample sizes are substantial with high class-imbalance. It was sufficient to use linear kernels to achieve high classification performance with the Word2vec representation of the features of the classified texts - the results obtained with non-linear kernels are similar. In the case of binary feature representation, the same performance level or

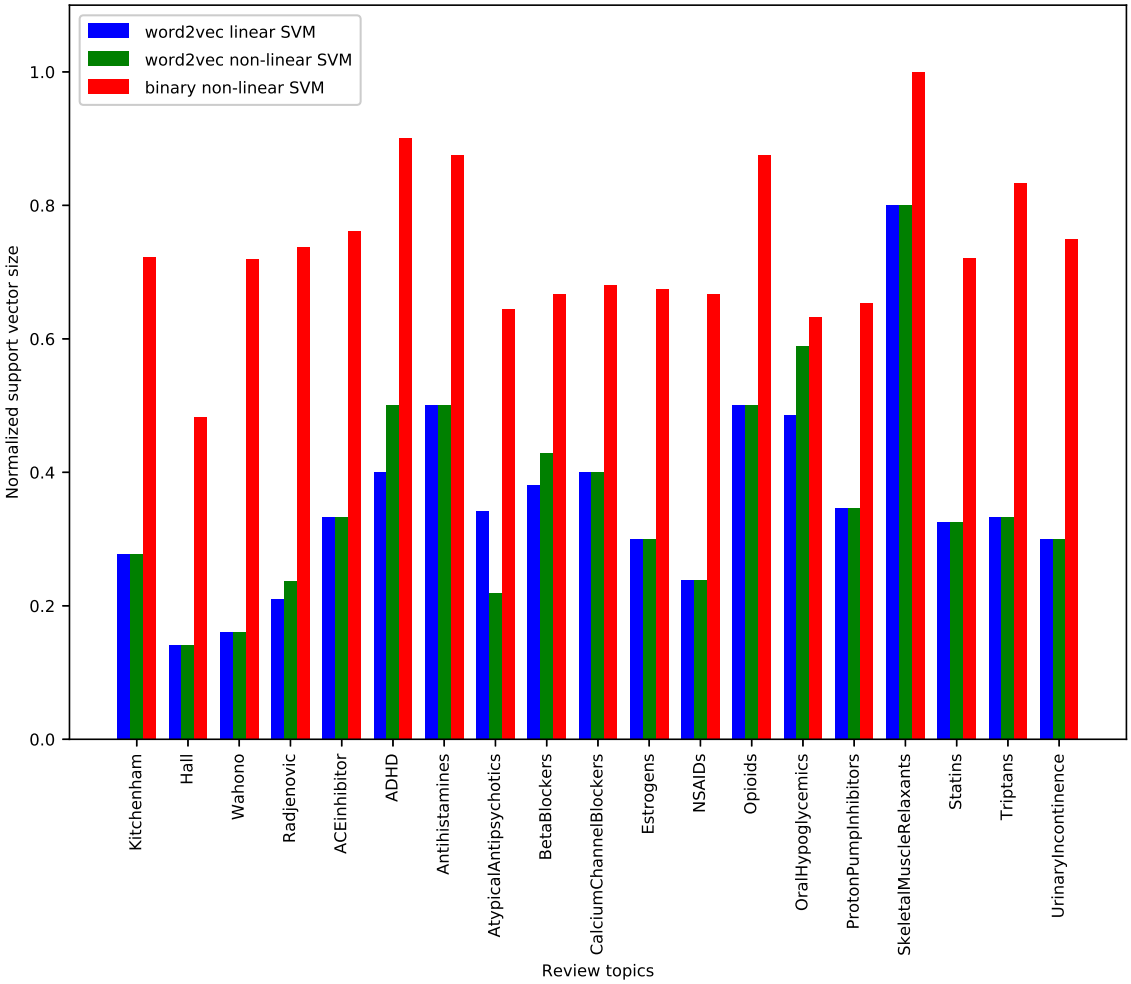

Figure 5.3: Normalized size ratio of positive SVs

higher can be achieved only with non-linear kernels. This indicates that the separator of the two classes in the data has a more non-linear nature in the case of text representation using binary features. Tables 5.4 and 5.5 show that models generated using Word2vec features have comparable performances and in some cases the number of SVs for the linear kernels were significantly smaller than for non-linear kernels and in other cases the reverse was true. Tables 5.4, 5.5 and 5.6 show that models using the Word2vec and binary features could have similar performances, however the number of SVs were significantly higher for models working with binary features and non-linear kernels. The bar charts of the average number of negative and positive samples used as SVs with their respective error margins by the different models are displayed in Figures 5.4 and 5.5.

Two approaches for feature representation binary and average word vector were explored. The binary features were better modelled by SVMs with non-linear kernels, while the average word vector features have a possibility of being modelled by either the linear or the non-linear kernel were explore machines for each review. With this

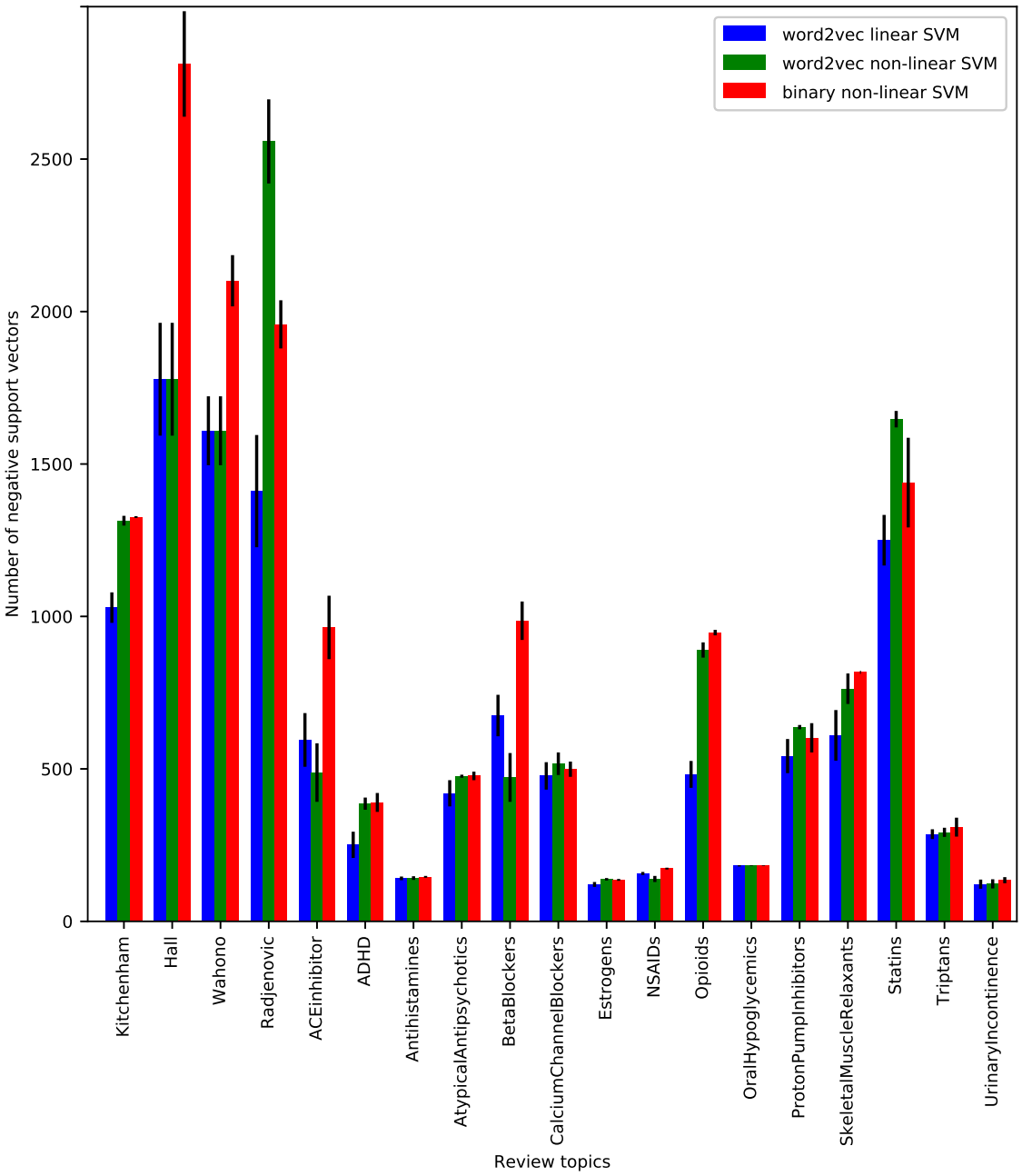

Figure 5.4: Negative samples used as training and SVs

approach, three models with similar performance for each of the reviews were studied. Taking complexity into account and the principles of model selection discussed in Section 5.2, the preferred SVM models are the ones that use linear kernels in most of the cases. The linear kernel based models were less complex than the non-linear ones in terms of the number of SVs in seven or 13 cases (as the case may be) and have comparable data description performance (i.e. recall and precision) with the models with non-linear kernels. This study has shown that it is possible for a model with high performance to have high complexity. Therefore, it is not sufficient to report

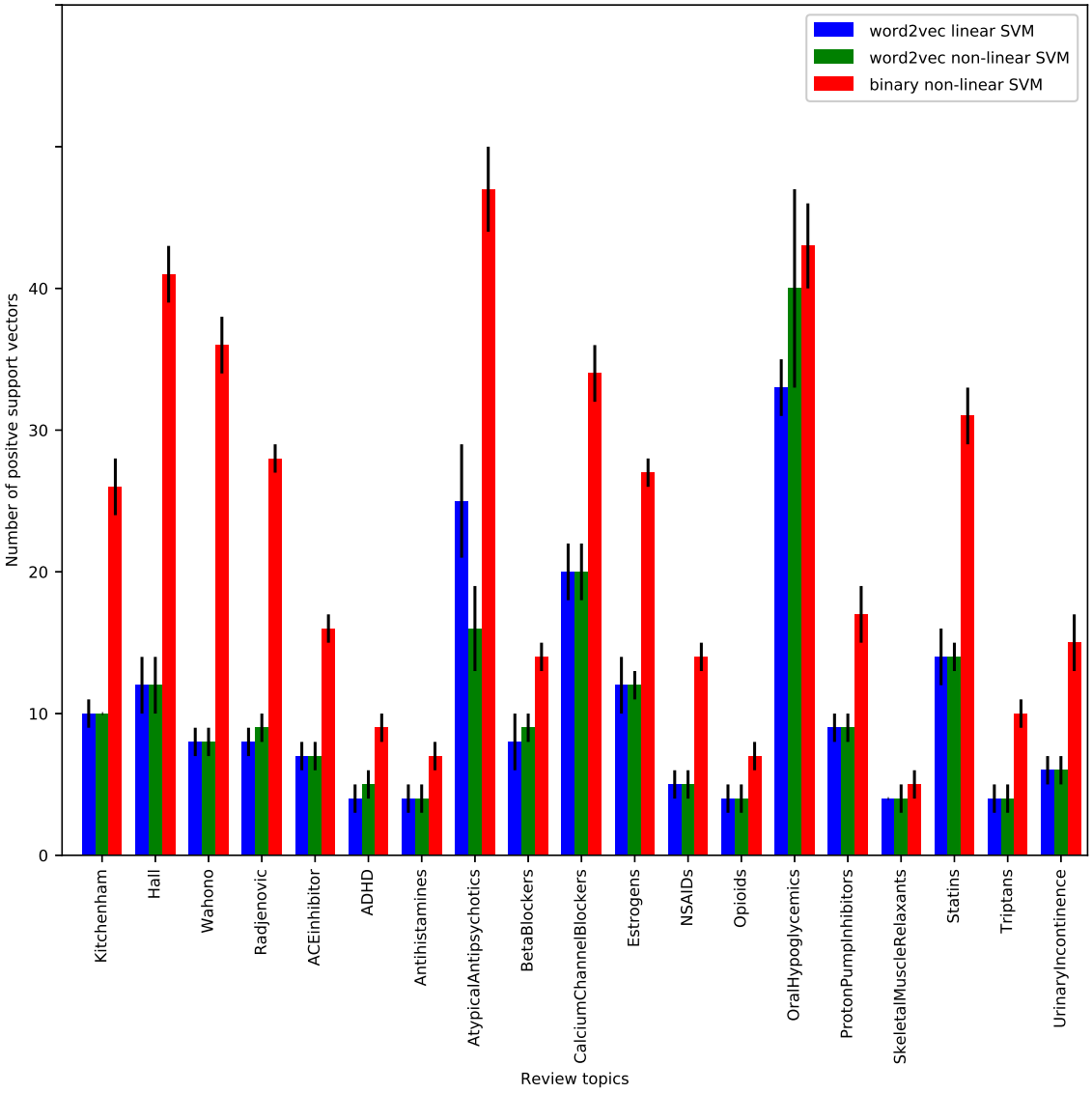

Figure 5.5: Positive samples used as training and SVs

only the performance results without complexity information. The study has also shown that using different feature types could lead to different level of complexity in models from the same data. Therefore, it is beneficial to explore and report a range of model parameters and particularly feature representations before optimisation of machine learning based models used for text classification.

Of course, results would be different for datasets with different class distributions and size. However, currently available SR datasets are relatively small for effective robust model learning and are typically highly imbalanced. The larger SE datasets showed more robust learning by using the smallest SV ratio particularly in the Word2vec feature based models. This is indicative of the role of dataset size in improved model learning, better generalization and invariably reduced complexity of models. Therefore, in order to reduce the number of SVs in a SVM model, an alternat-

ive way may be to increase the dataset size using the over-sampling method (Wallace, Trikalinos, et al., 2010; A. M. Cohen, 2006). This method had been used in previous studies, but the number of SVs were not reported; which was the main issue being addressed by this study reporting of more details for improved third party 'under the hood' understanding of the models' quality and performances.

Providing such information will not only be fulfilling a basic scientific requirement to aid comprehensibility and reproducibility but also help independent researchers who may be relying on or researching into such models see beyond performance metrics. They may then be able to answer the *'why'* question of *'what'* they observe which may lead to the process of *'how'* it can be improved.

5.9 Summary

The study reported in this chapter has investigated the possibility of complexity and validity concerns surrounding the TM models currently being proposed for the automatic screening of citations in systematic reviews. The study consequently made a case on the need for study reports in this field to contain alongside classification performance results, model related information revealing the complexity and validity of such models. This will first be in compliance with the scientific knowledge requirement with this type of research and in addition, give independent researchers a better understanding of the model and more grounds for reproducibility, comparability and improvement.

Complexity is a phenomenon common to all models, however, the specific complexity measure differ from model to model. In some models, it may be determined by the number of estimators or the count of non-zero points; in a tree algorithm it may be measured by the number depth of the tree and the number of leaves. This study only illustrated with the SVMs. The complexity information will indicate whether a model had actually learnt from or fit to the noise of its underlying data. The results of the study has shown that it is possible for a model to have good performance but still have inherent complexity and validity concerns. It has thus justified the need to provide corresponding information on these key concepts as a scientific need, for quality assessment and for the purpose of further research.

The study experimented with 19 reviews datasets - four from software engineering and 15 from the medical research. Support vector machines were developed based on binary features and Word2vec features. Represented with the binary features, the datasets were found not to be linearly separable by the SVM. However, this changed with the Word2vec representation. On the average, the Word2vec features based models with linear kernels also used fewer SVs than their non-linear kernel counterpart and the binary features based models with non-linear kernels.

The results of the study have shown that it is possible for a model to exhibit good

performance but have inherent complexity and validity concerns. Thus, it justified the need for the provision of complexity related information specific to each model in reports to facilitate understanding, reproduction and extension of the studies. The findings in this study have also provided the basis for investigation into how the performance observed can be improved without increasing the models' complexities. This idea will be explored in Chapter 6 where it will be investigated whether using bibliography data can improve the quality of input to compensate for the relatively small data sizes and class imbalance between the relevant and irrelevant articles and eventually improve the classification performance of the models.

CHAPTER 6

Feature Enrichment

The complexity study reported in Chapter 5 identified the risk of high complexity in existing models being proposed for automatic CS in SRs. This is potentially as a result of small datasets and thus motivated the need for study reports to include information that may reveal the level of complexity of the proposed models. This chapter investigates the effect of using bibliography information to try to improve the performance of models without increasing their complexity. The study uses the same dataset and follows the same process as in Section 5.4 but in this case two sets of features are prepared for each dataset, one as in Section 5.4 and the second with bibliography features added. The performance and complexity (in terms of the number of SVs used) of models from each corresponding set are compared and a t-test is used to investigate any actual difference in the complexities based on the number of SVs. The findings in this study indicated that the inclusion of bibliography data holds the potential to improve the performance of the automatic CS models. Though, no definitive pattern could be drawn on the effect of the bibliography information on the performance of the CS models in the DERP datasets and one (the smallest) of the SE datasets, it is however clear that the inclusion of the bibliography data is more likely to improve or sustain the model performance than impair it.

6.1 Introduction

The study reported in this chapter investigated the effect of adding bibliography data to the articles' titles and abstracts that was used in the complexity study reported in Chapter 5. This study used the same datasets and classification models as used in the complexity study (Chapter 5). In addition, the bibliography data for each candidate article was downloaded, cleaned and added as input data for the TM process. The pre-processing and feature representation was the same as described in Sections 5.4.2 and 5.4.3 respectively. The feature selection stage however explored the use of (new) variable α values for the χ^2 method in contrast to the uniform 5%

value recommended in (A. M. Cohen et al., 2006) for the DERP datasets. This is particularly motivated from the fact that no such benchmark existed for the SE datasets. It also served as an opportunity to explore other values for the DERP datasets. The goal of feature selection was to reduce the feature vector dimension by selecting the smallest possible top features that resulted in the highest performance of the model been built.

The rest of this chapter is structured as follows: An overview of the class imbalance problem is presented in Section 6.2. The conduct of the study is presented in Section 6.3 with results in Section 6.4. The limitations to the study are highlighted in Section 6.5 followed by discussion in Section 6.6 and a summary to the chapter in Section 6.7.

6.2 Mitigating class imbalance effect

As discussed in Section 2.1, supervised ML algorithms typically learn patterns underlying the example data and project the knowledge to predict similarity or otherwise of new data to the learned example (Murphy, 2012). A major problem in using these algorithms for classification purposes in automatic CS is the small number of relevant (positive class) examples to learn from compared to the number of irrelevant (negative class) examples. The proportion of relevant to irrelevant class examples is typically 1%-5% of the total data size. This situation is referred to as class imbalance (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). Imbalanced data classes impair the performance of classification models in ML.

Owing to the highly imbalanced nature of the data classes in SRs and its consequent effect, researchers working on automating the CS stage continue to explore ways to make up for the shortage of relevant class examples.

Some of the methods the community have explored to address this situation are:

a Cost Assignment: Assignment of different costs or weights to training samples (Domingos, 1999a).

b Data resampling: The repeated sampling of the original data either by over-sampling or under-sampling (Japkowicz, 2000; Kubat & Matwin, 1997):

- Over-sampling: This involves including repeated or multiple instances of the minority class samples to make up for its under-representation during training.
- Under-sampling: The process of under-sampling involves reducing the samples of the majority class to create a ‘reasonable’ representation proportion between the majority and the minority class samples.

c SMOTing: Using the synthetic data produced from the Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2002). The SMOTE combines both the over-sampling and under-sampling techniques to produce new data samples of both classes in the proportion specified by the user.

d Feature enrichment: An approach used in text classification to improve model performance by adding other possibly useful information, sometimes from (external) sources (Hu et al., 2008; P. Wang & Domeniconi, 2008). In the context of automatic CS with TM techniques it can be said to be the inclusion of other data beyond title and the abstract (that would have been ordinarily assessed by a human) e.g. pre-trained embedding (sometimes from external sources), keywords, subject classification data, cited articles etc. to provide more information that could potentially strengthen the probability of identifying similarities or differences between articles (P. Wang & Domeniconi, 2008; Hu et al., 2008; Khabsa et al., 2016).

Despite these efforts, there is yet to be an acceptable solution to the problem of stemming the effect of class imbalance in building TM models to automatically screen citations. The use of bibliography information to enrich the dataset is explored in the study reported in this chapter.

Introducing external data as a way of enriching the base data is one way the community continue to explore to tackle the effect of class imbalance on model learning. This approach attempts to leverage the machine speed and power by increasing the basic textual input data (titles and abstracts) to provide (possibly) more information from each article that could further show which ones are related or not.

One of the earliest attempts at feature enrichment in automatic CS research was the addition of Medical Subject Heading (MeSH) and the MEDLINE publication type data to the abstract and title (A. M. Cohen et al., 2006). A number of studies have used a similar approach. In (Bekhuis & Demner-Fushman, 2010) the authors mention using metadata alongside title and abstract and the authors of (A. M. Cohen, 2008) mention using MeSH and Natural Language Processing (NLP) features. Katia R Felizardo et al. (2012) used the mapping of citations to the contents that contain them to create article clusters for the identification of relevant citations.

Khabsa et al. used co-citation and clustering features to improve the feature quality of 15 SRs dataset in (Khabsa et al., 2016). For co-citation they worked on the assumption that if two articles are cited together in a third article, then both articles are likely to be on a similar subject. Therefore, either of the two articles that was not initially included in the dataset to be classified is retrieved and included as a positive sample. They further used the brown clustering algorithm (Brown, Desouza, Mercer, Pietra, & Lai, 1992) to create word clusters containing related words. With the cluster, each word is represented with a code that refers to a cluster of similar

words which might have appeared in the training corpus.

6.3 Feature enrichment study

This study followed and built on the process described in Section 5.4. Any differences are highlighted in this section. The goal of this study is to investigate the effect of using the bibliography data to enrich the input data - title and abstract, in automatic CS with TM techniques on the overall performance of the classification models.

6.3.1 Data retrieval

As indicated in the introduction to this section, the conduct of this study derive from the approach described in Section 5.4. This section therefore is built upon the data retrieval process described in Section 5.4.1. However, in this study, two sets of data were prepared for each of the 19 reviews used. The first set - TiAbs(MeSH), contained title and abstract with an additional MeSH feature for the 15 clinical review datasets. The second set - TiAbs(MeSH)Ref, contained the first set and the full reference list for each candidate article (where available, accessible and retrievable).

Based on the data retrieval process described in Section 5.4.1, the TiAbs(MeSH) set were readily available. Custom scripts were written to automatically search and retrieve the reference list to make up set two. The process followed to retrieve the reference list for each candidate article in each of the two reviews set (SE and EPC) used is described below:

i) SE dataset

   a) the full article link provided for the four SE reviews were used to search for the articles in the publishers' website.
   b) where possible and available the bibliography for each candidate article were automatically extracted.

ii) EPC dataset

   a) the PMIDs provided in the supporting material to (A. M. Cohen et al., 2006) were used to automatically search the pubmed database[1] for available information on each candidate article content of each of the candidate.
   b) the information of the publisher(s) providing access to the full content of each article were scraped from the information.

[1] https://www.ncbi.nlm.nih.gov/pubmed/

c) this information (if one was found) was used to traverse the publishers' site for an attempt to retrieve the desired reference list for each of the candidate articles

The retrieved reference texts were initially cleaned of Hypertext Markup Language (HTML) tags and any Universal Resource Locator (URL) information before being merged with TiAbs(MeSH) data to form the set two data - TiAbs(MeSh)Ref.

6.3.2 Feature selection

The top features (selected after ranking) to reduce the dimension of resulting feature vector were determined using the χ^2 method as was described in Section 5.4.4. However, instead of selecting the top 5% features ($\alpha = 0.05$, refer to Section 2.2.4.1 for discussion on χ^2 and α in feature selection context) as used in (A. M. Cohen et al., 2006), a set of fresh values (between 1% and 50%) were explored mainly because no study has established such benchmark for the SE datasets and also to investigate other possible values for the DERP datasets. This goal was achieved by setting the α-value in the sense of optimality defined in this study.

The rest of the steps not described are the same as was described in Section 5.4

6.4 Results

The findings from the study are presented in this section.

6.4.1 Data retrieval

The number of candidate articles for each review and their class distribution is shown in Table 5.1. The number of references found per review is shown in Table 6.1 with detailed distribution according to the classes in Table 6.2.

6.4.2 Feature representation

As was discussed in Section 5.4.3, the binary features produced good results only with the non-linear kernels of the SVM. The Word2vec feature representation on the other hand showed comparable performance with both the linear and non-linear kernels of the SVM. Therefore, models were built from the *binary-non-linear, Word2vec-linear and Word2vec-non-linear* feature representation-SVM kernel combinations.

6.4.3 Dimensionality reduction

Setting $\alpha = 5\%$ for χ^2 in Section 5.4.4 did not result in the exact values as given in (A. M. Cohen et al., 2006) for the DERP datasets, this in addition to the fact that

Table 6.1: Number of references retrieved per study

Review	Not found	Found
Kitchenham	60	1644
Hall	408	8503
Wahono	313	6689
Radjenovic	347	5653
ACEinhibitor	1533	1011
ADHD	484	367
Antihistamines	192	118
AtypicalAntipsychotics	707	413
BetaBlockers	1182	890
Calcium...Blockers	770	448
Estrogens	206	162
NSAIDs	223	170
Opioids	1123	791
OralHypoglycemics	295	205
Proton..Inhibitors	802	531
Skeletal...Relaxants	1079	564
Statins	2040	1425
Triptans	367	304
UrinaryIncontinence	178	149

Table 6.2: Class distribution of retrieved references

Review	Positive class		Negative class	
	Not found	Found	Not found	Found
Kitchenham	2	43	58	1601
Hall	1	105	407	8398
Wahono	2	60	311	6629
Radjenovic	1	47	346	5616
ACEinhibitor	24	27	1509	994
ADHD	7	13	477	354
Antihistamines	12	4	180	114
AtypicalAntipsychotics	77	69	630	344
BetaBlockers	15	27	1167	863
Calcium...Blockers	44	56	726	392
Estrogens	41	39	165	187
NSAIDs	20	21	203	149
Opioids	8	7	1116	784
OralHypoglycemics	68	68	230	138
Proton...Inhibitors	23	28	779	503
Skeletal...Relaxants	5	4	1074	560
Statins	47	38	1993	1387
Triptans	7	17	360	287
UrinaryIncontinence	18	22	160	127

there exists no known similar benchmark on the SE datasets, informed the decision to explore different values to confirm which value would result in a reduced vector dimension with highest 'acceptable recall' value. 'Acceptable recall' value in this context implied the highest possible value of recall where the model still exhibited some discriminatory power over the dataset. It was interesting to find that better recall values can be obtained for the datasets at values of α other than 0.05.

Starting with the binary features of the TiABs(MeSH) data, it was found that each of the datasets performed best for different α-values with the χ^2 method used. However, the majority of the datasets seemed to start recording high ($\geq\ 90\%$) recall performance at around $\alpha = 5\%$ top percentile value. The performance around the 5% percentile is consistent with the findings reported in (A. M. Cohen et al., 2006). The different alpha values used and their corresponding feature size for the TiAbs(MeSH) data are presented in Table 6.3a. The TiAbs(MeSH) data recall performance results are used as a benchmark to search for the appropriate reduced dimension of the TiAbs(MeSH)Ref data that will produce equal, close enough or better recall performance than was initially observed in the TiAbs(MeSH) feature SVM models. The resulting feature sizes and their corresponding α values are shown in Table 6.3b.

6.4.4 Model assessment

6.4.4.1 Performance measures

With the binary feature, the TiAbs(MeSH)Ref data exhibited better recall values than the TiAbs(MeSH) data in 12 reviews, equal in four and worse in two. The models could not produce any useful results for the 'SKeletalMuscleRelaxants' data despite the fact that it is larger than some other datasets in the collection. This might be because it has the smallest number of positive candidates (9 compared to the negative class size of 1634, see Table 6.2).

Tables 6.5a and 6.5b show the results of the SVM linear models for the TiAbs(MeSH) and TiAbs(MeSH)Ref Word2vec features respectively. The tables show that the TiAbs(MeSH)Ref data has higher recall in nine reviews, lower in seven reviews and equal to the TiAbs(MeSH) data in three reviews.

With the Word2vec feature representation and SVM non-linear kernels, the TiAbs(MeSH)Ref data showed higher recall in six reviews (Table 6.6b), lower recall values in nine reviews and equal recall values in four reviews compared to the TiAbs(MeSH) data (Table 6.6a).

Considering the MCC, which is a measure that takes all the four basic model performance measures (TN, FN, TP and FP) into account, the TiAbs(MeSH)Ref data recorded higher values in 11 of the 19 reviews compared to the TiAbs(MeSH) data with the non-linear kernel of the SVM and Word2vec feature (Tables 6.6b and 6.6a). For the binary feature the TiAbs(MeSH) feature (Table 6.4a) recorded

Table 6.3: χ^2 selected top features

(a) TiAbs(MeSH) data

Reviews	Initial size	α value	final size
Kitchenham	5730	4	227
Hall	11834	8	947
Wahono	11137	6	668
Radjenovic	10165	5	508
ACEinhibitor	4933	5	246
ADHD	3017	4	122
Antihistamines	1570	2	31
AtypicalAntipsychotics	3237	3	98
BetaBlockers	4724	4	192
Calcium...Blockers	3462	4	138
Estrogens	1861	18	339
NSAIDs	1790	21	376
Opioids	4661	1	46
OralHypoglycemics	2112	10	211
Proton...Inhibitors	3299	5	165
Skeletal...Relaxants	4826	1	48
Statins	6150	5	308
Triptans	2372	5	118
UrinaryIncontinence	1691	30	5075

(b) TiAbs(MeSH)Ref data

Reviews	Initial size	α value	final size
Kitchenham	20095	3.8	763
Hall	44302	5	2215
Wahono	41800	2	836
Radjenovic	33929	1	339
ACEinhibitor	3808	1.8	248
ADHD	7680	4	307
Antihistamines	2983	3.5	105
AtypicalAntipsychotics	7920	3	236
BetaBlockers	14510	4	580
Calcium...Blockers	90332	6	542
Estrogens	4780	10	478
NSAIDs	4457	21	936
Opioids	12034	0.9	116
OralHypoglycemics	5050	8	404
Proton...Inhibitors	8251	2.5	206
Skeletal...Relaxants	11723	1	118
Statins	19454	3	584
Triptans	4969	3	150
UrinaryIncontinence	3634	15	545

Table 6.4: Binary feature non-linear kernel

(a) TiAbs(MeSH) data

Reviews	Mean Performance					Support vectors		Configuration
	precision	recall	accuracy	WSS	MCC	neg	pos	parameters[2]
Kitchenham	0.04 ± 0.01	0.93 ± 0.04	0.44 ± 0.16	0.35 ± 0.11	0.12 ± 0.03	1352 ± 1	25 ± 1	rbf, 1.0
Hall	0.33 ± 0.03	0.93 ± 0.04	0.98 ± 0.00	0.90 ± 0.04	0.55 ± 0.03	2238 ± 152	48 ± 2	sigmoid, 1.0
Wahono	0.19 ± 0.02	0.91 ± 0.09	0.97 ± 0.00	0.86 ± 0.09	0.41 ± 0.05	1947 ± 84	38 ± 2	sigmoid, 1.0
Radjenovic	0.13 ± 0.03	0.77 ± 0.11	0.96 ± 0.01	0.72 ± 0.11	0.31 ± 0.05	1961 ± 96	28 ± 1	sigmoid, 1.0
ACEinhibitor	0.14 ± 0.02	0.84 ± 0.06	0.91 ± 0.02	0.74 ± 0.05	0.32 ± 0.03	898 ± 58	15 ± 1	sigmoid, 1.0, .001
ADHD	0.13 ± 0.02	0.95 ± 0.05	0.85 ± 0.02	0.78 ± 0.05	0.32 ± 0.03	316 ± 33	10 ± 0	rbf, 1.0, .001
Antihistamines	0.06 ± 0.02	0.59 ± 0.37	0.46 ± 0.35	0.04 ± 0.07	0.02 ± 0.04	139 ± 16	8 ± 0	rbf, 10, .001
AtypicalAntipsychotics	0.22 ± 0.02	0.81 ± 0.04	0.59 ± 0.06	0.32 ± 0.04	0.25 ± 0.03	465 ± 16	39 ± 2	rbf, 1.0
BetaBlockers	0.05 ± 0.02	0.91 ± 0.10	0.63 ± 0.17	0.52 ± 0.11	0.17 ± 0.04	1009 ± 15	14 ± 2	sigmoid, 1.0, .001
Calcium...Blockers	0.23 ± 0.03	0.77 ± 0.07	0.76 ± 0.04	0.49 ± 0.06	0.33 ± 0.04	441 ± 30	29 ± 2	rbf, 1.0
Estrogens	0.36 ± 0.03	0.97 ± 0.03	0.61 ± 0.05	0.38 ± 0.04	0.41 ± 0.04	141 ± 2	28 ± 2	rbf, 1.0
NSAIDs	0.33 ± 0.04	0.94 ± 0.06	0.79 ± 0.03	0.64 ± 0.04	0.48 ± 0.04	165 ± 9	15 ± 2	rbf, 10, .0001
Opioids	0.06 ± 0.05	0.81 ± 0.22	0.55 ± 0.45	0.36 ± 0.32	0.13 ± 0.12	904 ± 136	6 ± 1	sigmoid, 1.0, .001
OralHypoglycemics	0.29 ± 0.02	0.97 ± 0.05	0.33 ± 0.07	0.05 ± 0.05	0.09 ± 0.07	184 ± 0	40 ± 4	sigmoid, 1.0, .001
Proton...Inhibitors	0.08 ± 0.02	0.88 ± 0.08	0.61 ± 0.11	0.46 ± 0.07	0.19 ± 0.03	624 ± 30	16 ± 1	rbf, 1.0, .001
Skeletal...Relaxants	0.00 ± 0.00	0.00 ± 0.00	0.99 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	735 ± 103	4 ± 0	rbf, 1.0, .001
Statins	0.06 ± 0.01	0.87 ± 0.08	0.67 ± 0.10	0.52 ± 0.04	0.18 ± 0.02	1584 ± 114	27 ± 2	sigmoid, 1.0, .001
Triptans	0.11 ± 0.04	0.81 ± 0.14	0.68 ± 0.21	0.47 ± 0.11	0.22 ± 0.06	307 ± 27	10 ± 1	rbf, 1.0, .001
UrinaryIncontinence	0.25 ± 0.01	0.88 ± 0.12	0.55 ± 0.25	0.35 ± 0.18	0.28 ± 0.15	143 ± 2	17 ± 1	sigmoid, 1.0, auto

[2]Parameter — kernel, C, gamma[a]

[a]In this and similar following tables, the default 'auto' value is used for gamma where not explicitly stated. Note that the value used by the algorithm in such a case remains unknown

(b) TiAbs(MeSH)Ref data

Reviews	Mean Performance					Support vectors		Configuration
	precision	recall	accuracy	WSS	MCC	neg	pos	parameters[3]
Kitchenham	0.03 ± 0.01	0.94 ± 0.0	0.24 ± 0.21	0.16 ± 0.16	0.06 ± 0.05	1327 ± 1	28 ± 1	rbf, 1.0
Hall	0.36 ± 0.04	0.93 ± 0.0	0.98 ± 0.00	0.90 ± 0.08	0.24 ± 0.57	2002 ± 244	37 ± 2	sigmoid, 1.0
Wahono	0.13 ± 0.04	0.94 ± 0.00	0.94 ± 0.02	0.87 ± 0.06	0.33 ± 0.06	2408 ± 491	32 ± 3	sigmoid, 1.0
Radjenovic	0.09 ± 0.01	0.85 ± 0.01	0.93 ± 0.01	0.78 ± 0.11	0.26 ± 0.03	1642 ± 237	15 ± 1	sigmoid, 1.0
ACEinhibitor	0.06 ± 0.02	0.84 ± 0.11	0.70 ± 0.24	0.53 ± 0.19	0.17 ± 0.06	1250 ± 3	13 ± 2	sigmoid, 1.0, .001
ADHD	0.12 ± 0.03	0.91 ± 0.08	0.82 ± 0.06	0.71 ± 0.04	0.28 ± 0.04	348 ± 50	8 ± 1	rbf, 1.0, .001
Antihistamines	0.06 ± 0.02	0.65 ± 0.33	0.42 ± 0.34	0.05 ± 0.1	0.03 ± 0.06	144 ± 10	8 ± 0	rbf, 10, .001
AtypicalAntipsychotics	0.15 ± 0.03	0.95 ± 0.07	0.27 ± 0.17	0.10 ± 0.13	0.09 ± 0.1	487 ± 1	42 ± 3	rbf, 1.0
BetaBlockers	0.04 ± 0.02	0.90 ± 0.14	0.46 ± 0.27	0.34 ± 0.19	0.11 ± 0.05	1015 ± 0	17 ± 2	sigmoid, 1.0, .001
Calcium...Blockers	0.12 ± 0.03	0.92 ± 0.08	0.41 ± 0.17	0.26 ± 0.11	0.17 ± 0.06	556 ± 7	37 ± 2	rbf, 1.0
Estrogens	0.23 ± 0.01	0.99 ± 0.01	0.29 ± 0.05	0.07 ± 0.05	0.13 ± 0.05	143 ± 2	31 ± 2	rbf, 1.0
NSAIDs	0.35 ± 0.02	0.96 ± 0.04	0.81 ± 0.02	0.67 ± 0.04	0.50 ± 0.03	160 ± 4	18 ± 0	rbf, 10, .0001
Opioids	0.06 ± 0.08	0.83 ± 0.23	0.38 ± 0.46	0.21 ± 0.27	0.1 ± 0.14	909 ± 124	6 ± 1	sigmoid, 1.0, .001
OralHypoglycemics	0.28 ± 0.02	0.97 ± 0.05	0.32 ± 0.03	0.05 ± 0.05	0.06 ± 0.07	181 ± 1	48 ± 4	sigmoid, 1.0, .001
Proton...Inhibitors	0.08 ± 0.03	0.90 ± 0.09	0.48 ± 0.26	0.35 ± 0.19	0.15 ± 0.08	633 ± 16	16 ± 2	rbf, 1.0, .001
Skeletal...Relaxants	0.00 ± 0.00	0.1 ± 0.03	0.89 ± 0.3	0.00 ± 0.00	0.00 ± 0.00	583 ± 191	4 ± 0	rbf, 1.0, .001
Statins	0.06 ± 0.01	0.87 ± 0.08	0.66 ± 0.09	0.52 ± 0.06	0.17 ± 0.02	1602 ± 108	28 ± 3	sigmoid, 1.0, .001
Triptans	0.09 ± 0.05	0.86 ± 0.15	0.47 ± 0.36	0.30 ± 0.25	0.14 ± 0.12	318 ± 18	9 ± 1	rbf, 1.0, .001
UrinaryIncontinence	0.21 ± 0.05	0.90 ± 0.07	0.55 ± 0.15	0.35 ± 0.12	0.27 ± 0.08	143 ± 2	15 ± 1	sigmoid, 1.0

[3]Parameter — kernel, C, gamma

better MCC values than the TiAbs(MeSH)Ref feature (Table 6.4b) in all the reviews. With the linear SVM kernel and Word2vec feature however, the TiAbs(MeSH)Ref data (Table 6.5b) showed higher MCC values than the TiAbs(MeSH) data (Table 6.5b) in nine reviews and equal values in one review.

The TiAbs(MeSH)Ref data appeared to be saving more work over random sampling in 15 out of the 19 reviews (see WSS in Table 6.6b and 6.6a). Given the binary fea-

Table 6.5: Word2vec feature with linear SVM kernel

(a) TiAbs(MeSH) data

Reviews	Mean Performance					Support vectors		Configuration
	precision	recall	accuracy	WSS	MCC	neg	pos	parameters[4]
kitchenham	0.06 ± 0.01	0.91 ± 0.08	0.59 ± 0.04	0.48 ± 0.08	0.16 ± 0.02	957.0 ± 79.0	11.0 ± 1.0	100
Hall	0.11 ± 0.01	0.97 ± 0.04	0.91 ± 0.01	0.86 ± 0.03	0.31 ± 0.02	1732.0 ± 242.0	12.0 ± 1.0	1
Wahono	0.07 ± 0.01	0.96 ± 0.05	0.89 ± 0.01	0.84 ± 0.05	0.25 ± 0.02	1533.0 ± 119.0	9.0 ± 1.0	1
Radjenovic	0.05 ± 0.01	0.92 ± 0.1	0.87 ± 0.02	0.78 ± 0.08	0.2 ± 0.02	1442.0 ± 185.0	10.0 ± 1.0	1
ACEInhibitors	0.08 ± 0.02	0.96 ± 0.04	0.8 ± 0.05	0.74 ± 0.04	0.24 ± 0.03	590.0 ± 101.0	7.0 ± 1.0	1
ADHD	0.08 ± 0.0	0.96 ± 0.08	0.75 ± 0.02	0.68 ± 0.06	0.24 ± 0.02	252.0 ± 33.0	4.0 ± 1.0	1
Antihistamines	0.06 ± 0.0	0.9 ± 0.11	0.21 ± 0.11	0.07 ± 0.04	0.04 ± 0.03	140.0 ± 7.0	5.0 ± 1.0	40
AtypicalAntipsychotics	0.18 ± 0.01	0.9 ± 0.04	0.45 ± 0.05	0.24 ± 0.03	0.2 ± 0.02	417.0 ± 13.0	28.0 ± 1.0	1000
BetaBlockers	0.05 ± 0.0	0.91 ± 0.06	0.64 ± 0.04	0.53 ± 0.03	0.16 ± 0.01	683.0 ± 58.0	8.0 ± 1.0	1
Calcium...Blockers	0.13 ± 0.01	0.92 ± 0.04	0.47 ± 0.04	0.32 ± 0.03	0.19 ± 0.02	461.0 ± 26.0	20.0 ± 1.0	100
Estrogens	0.3 ± 0.02	0.93 ± 0.04	0.52 ± 0.05	0.26 ± 0.04	0.3 ± 0.03	125.0 ± 8.0	12.0 ± 1.0	1000
NSAIDS	0.15 ± 0.01	1.0 ± 0.0	0.39 ± 0.03	0.28 ± 0.03	0.21 ± 0.02	158.0 ± 2.0	6.0 ± 0.0	1
Opiods	0.03 ± 0.01	0.8 ± 0.12	0.78 ± 0.06	0.57 ± 0.1	0.13 ± 0.03	469.0 ± 65.0	4.0 ± 1.0	1
OralHypoglycemics	0.28 ± 0.01	0.99 ± 0.01	0.3 ± 0.02	0.02 ± 0.02	0.07 ± 0.04	183.0 ± 1.0	34.0 ± 3.0	10000
Proton...Inhibitors	0.06 ± 0.01	0.94 ± 0.05	0.44 ± 0.09	0.35 ± 0.06	0.15 ± 0.02	545.0 ± 56.0	9.0 ± 1.0	1
Skeletal...Relaxants	0.01 ± 0.0	0.64 ± 0.28	0.55 ± 0.14	0.2 ± 0.21	0.03 ± 0.03	581.0 ± 136.0	4.0 ± 1.0	1
Statins	0.05 ± 0.01	0.93 ± 0.03	0.56 ± 0.05	0.46 ± 0.03	0.15 ± 0.01	1252.0 ± 73.0	15.0 ± 1.0	1
Triptans	0.06 ± 0.01	0.94 ± 0.12	0.45 ± 0.11	0.36 ± 0.07	0.15 ± 0.02	283.0 ± 27.0	4.0 ± 1.0	1
UrinaryIncontinence	0.2 ± 0.03	0.94 ± 0.07	0.5 ± 0.11	0.33 ± 0.08	0.26 ± 0.05	128.0 ± 16.0	5.0 ± 1.0	100

[4]Parameter — C

(b) TiAbs(MeSH)Ref data

Reviews	Mean Performance					Support vectors		Configuration
	precision	recall	accuracy	WSS	MCC	neg	pos	parameters[5]
Kitchenham	0.04 ± 0.01	0.88 ± 0.12	0.44 ± 0.13	0.3 ± 0.06	0.1 ± 0.02	1204.0 ± 112.0	14.0 ± 2.0	1
Hall	0.16 ± 0.01	0.97 ± 0.03	0.94 ± 0.01	0.9 ± 0.03	0.38 ± 0.02	1492.0 ± 250.0	12.0 ± 1.0	1
Wahono	0.11 ± 0.01	0.98 ± 0.03	0.93 ± 0.01	0.9 ± 0.02	0.31 ± 0.01	1255.0 ± 150.0	9.0 ± 1.0	1
Radjenovic	0.07 ± 0.01	0.96 ± 0.07	0.9 ± 0.01	0.85 ± 0.05	0.25 ± 0.01	1197.0 ± 123.0	8.0 ± 2.0	1
ACEInhibitors	0.08 ± 0.02	0.91 ± 0.07	0.8 ± 0.06	0.7 ± 0.07	0.23 ± 0.04	607.0 ± 123.0	7.0 ± 1.0	1
ADHD	0.09 ± 0.01	0.97 ± 0.05	0.75 ± 0.03	0.7 ± 0.03	0.25 ± 0.01	236.0 ± 33.0	4.0 ± 1.0	1
Antihistamines	0.05 ± 0.0	0.84 ± 0.19	0.2 ± 0.19	0.0 ± 0.05	0.0 ± 0.04	136.0 ± 22.0	5.0 ± 1.0	10
AtypicalAntipsychotics	0.2 ± 0.01	0.83 ± 0.05	0.53 ± 0.03	0.28 ± 0.02	0.22 ± 0.02	359.0 ± 19.0	33.0 ± 3.0	1000
BetaBlockers	0.04 ± 0.01	0.86 ± 0.05	0.58 ± 0.07	0.43 ± 0.06	0.13 ± 0.02	793.0 ± 74.0	10.0 ± 1.0	1
Calcium...Blockers	0.13 ± 0.01	0.93 ± 0.03	0.48 ± 0.04	0.34 ± 0.04	0.21 ± 0.03	484.0 ± 19.0	21.0 ± 2.0	10
Estrogens	0.36 ± 0.02	0.94 ± 0.04	0.62 ± 0.04	0.38 ± 0.03	0.4 ± 0.03	104.0 ± 7.0	12.0 ± 1.0	100
NSAIDS	0.14 ± 0.01	1.0 ± 0.01	0.37 ± 0.06	0.26 ± 0.05	0.2 ± 0.03	168.0 ± 5.0	7.0 ± 1.0	1
Opiods	0.04 ± 0.0	0.82 ± 0.17	0.82 ± 0.03	0.64 ± 0.14	0.15 ± 0.02	424.0 ± 35.0	5.0 ± 1.0	1
OralHypoglycemics	0.33 ± 0.02	0.91 ± 0.03	0.47 ± 0.04	0.16 ± 0.03	0.23 ± 0.04	165.0 ± 8.0	30.0 ± 2.0	10000
Proton...Inhibitors	0.05 ± 0.0	0.92 ± 0.08	0.37 ± 0.08	0.27 ± 0.03	0.11 ± 0.01	576.0 ± 44.0	11.0 ± 2.0	1
Skeletal...Relaxants	0.0 ± 0.0	0.26 ± 0.23	0.69 ± 0.11	-0.06 ± 0.17	-0.01 ± 0.03	530.0 ± 111.0	4.0 ± 0.0	1
Statins	0.04 ± 0.0	0.94 ± 0.04	0.5 ± 0.04	0.42 ± 0.03	0.13 ± 0.01	1365.0 ± 65.0	16.0 ± 2.0	1
Triptans	0.06 ± 0.01	0.94 ± 0.08	0.47 ± 0.12	0.38 ± 0.07	0.15 ± 0.02	269.0 ± 37.0	6.0 ± 1.0	1
UrinaryIncontinence	0.17 ± 0.01	0.95 ± 0.09	0.44 ± 0.06	0.28 ± 0.05	0.23 ± 0.03	122.0 ± 13.0	7.0 ± 1.0	10

[5]Parameter — C

ture and SVM non-linear modelS the TiAbs(MeSH)Ref (Table 6.4a) has higher WSS value in four reviews and equal values in four. In Word2vec based linear kernel SVM models the TiAbs(MeSH)Ref data (Table 6.5b) has higher MCC values than the TiAbs(MeSH) data(Table 6.5a) in 10 of the reviews.

6.4.4.2 Complexity measures

The TiAbs(MeSH)Ref data used fewer support vectors than the TiAbs(MeSH) data across all the SE datasets except the Kitchenham dataset as shown in the number of support vectors reported in Tables 6.4b, 6.5b and 6.6b. There is variation across the rest of the datasets of the number of support vectors used for both sets of the data.

Table 6.6: Word2vec feature non-linear kernel

(a) TiAbs(MeSH) Features

Reviews	Mean Performance					Support vectors		Configuration
	precision	recall	accuracy	WSS	MCC	neg	pos	parameters[6]
Kitchenham	0.04 ± 0.01	0.98 ± 0.04	0.36 ± 0.13	0.32 ± 0.11	0.11 ± 0.03	1299.0 ± 28.0	11.0 ± 1.0	rbf, 1000, 0.001
Hall	0.11 ± 0.01	0.97 ± 0.04	0.91 ± 0.01	0.86 ± 0.03	0.31 ± 0.02	1732.0 ± 242.0	12.0 ± 1.0	sigmoid, 1000, 0.001
Wahono	0.07 ± 0.01	0.96 ± 0.05	0.89 ± 0.01	0.84 ± 0.05	0.25 ± 0.02	1533.0 ± 119.0	9.0 ± 1.0	sigmoid, 1000, 0.001
Radjenovic	0.03 ± 0.0	0.96 ± 0.07	0.76 ± 0.02	0.72 ± 0.06	0.15 ± 0.01	2560.0 ± 170.0	9.0 ± 1.0	sigmoid, 100, 0.001
ACEInhibitors	0.09 ± 0.02	0.92 ± 0.06	0.83 ± 0.05	0.74 ± 0.04	0.25 ± 0.03	486.0 ± 103.0	7.0 ± 1.0	rbf, 1000, 0.001
ADHD	0.09 ± 0.01	0.95 ± 0.08	0.76 ± 0.02	0.69 ± 0.06	0.24 ± 0.02	210.0 ± 28.0	4.0 ± 1.0	rbf, 1000, 0.001
Antihistamines	0.06 ± 0.0	0.92 ± 0.1	0.18 ± 0.11	0.06 ± 0.04	0.05 ± 0.02	141.0 ± 7.0	4.0 ± 1.0	sigmoid, 1000
AtypicalAntipsychotics	0.15 ± 0.01	0.96 ± 0.03	0.29 ± 0.06	0.14 ± 0.04	0.14 ± 0.02	466.0 ± 12.0	24.0 ± 2.0	sigmoid, 10000, 0.001
BetaBlockers	0.07 ± 0.01	0.82 ± 0.05	0.76 ± 0.05	0.57 ± 0.07	0.19 ± 0.03	469.0 ± 54.0	9.0 ± 1.0	sigmoid, 1000
Calcium...Blockers	0.12 ± 0.01	0.93 ± 0.03	0.45 ± 0.04	0.31 ± 0.03	0.19 ± 0.02	472.0 ± 26.0	20.0 ± 1.0	sigmoid, 10000
Estrogens	0.24 ± 0.02	0.98 ± 0.03	0.32 ± 0.08	0.08 ± 0.06	0.13 ± 0.08	141.0 ± 4.0	12.0 ± 1.0	sigmoid, 10000
NSAIDS	0.17 ± 0.01	1.0 ± 0.01	0.51 ± 0.03	0.4 ± 0.03	0.28 ± 0.02	145.0 ± 4.0	5.0 ± 1.0	sigmoid, 1000
Opiods	0.02 ± 0.0	0.98 ± 0.05	0.6 ± 0.06	0.57 ± 0.06	0.1 ± 0.01	736.0 ± 41.0	4.0 ± 1.0	sigmoid, 10
OralHypoglycemics	0.27 ± 0.0	1.0 ± 0.0	0.27 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	184.0 ± 0.0	45.0 ± 5.0	sigmoid, 1000
Proton...Inhibitors	0.04 ± 0.0	0.98 ± 0.04	0.13 ± 0.09	0.07 ± 0.06	0.04 ± 0.03	639.0 ± 4.0	8.0 ± 1.0	sigmoid, 100, 0.001
Skeletal...Relaxants	0.01 ± 0.0	0.9 ± 0.2	0.31 ± 0.2	0.21 ± 0.08	0.04 ± 0.01	740.0 ± 103.0	4.0 ± 1.0	rbf, 100, 0.001
Statins	0.03 ± 0.0	0.98 ± 0.02	0.26 ± 0.06	0.22 ± 0.06	0.08 ± 0.02	1647.0 ± 27.0	14.0 ± 1.0	sigmoid, 100, 0.001
Triptans	0.06 ± 0.01	0.94 ± 0.12	0.42 ± 0.12	0.33 ± 0.07	0.14 ± 0.02	288.0 ± 25.0	5.0 ± 0.0	sigmoid, 100
UrinaryIncontinence	0.17 ± 0.04	0.93 ± 0.07	0.37 ± 0.2	0.19 ± 0.16	0.15 ± 0.12	131.0 ± 15.0	6.0 ± 1.0	rbf, 10000

[6]Parameter — kernel, C, gamma

(b) TiAbs(MeSH)Ref data

Reviews	Mean Performance					Support vectors		Configuration
	precision	recall	accuracy	WSS	MCC	neg	pos	parameters[7]
Kitchenham	0.03 ± 0.01	0.93 ± 0.09	0.21 ± 0.22	0.12 ± 0.15	0.04 ± 0.05	1308.0 ± 35.0	13.0 ± 2.0	rbf, 100, 0.001
Hall	0.16 ± 0.01	0.97 ± 0.03	0.94 ± 0.01	0.9 ± 0.03	0.38 ± 0.02	1492.0 ± 250.0	12.0 ± 1.0	sigmoid, 1000, 0.001
Wahono	0.11 ± 0.01	0.98 ± 0.03	0.93 ± 0.01	0.9 ± 0.02	0.31 ± 0.01	1255.0 ± 150.0	9.0 ± 1.0	sigmoid, 1000, 0.001
Radjenovic	0.09 ± 0.01	0.96 ± 0.07	0.92 ± 0.01	0.87 ± 0.05	0.27 ± 0.01	942.0 ± 103.0	10.0 ± 2.0	rbf, 1000, 0.001
ACEInhibitors	0.08 ± 0.02	0.91 ± 0.07	0.8 ± 0.06	0.7 ± 0.07	0.23 ± 0.04	607.0 ± 123.0	7.0 ± 1.0	sigmoid, 1000, 0.001
ADHD	0.09 ± 0.01	0.97 ± 0.05	0.75 ± 0.03	0.7 ± 0.03	0.25 ± 0.01	236.0 ± 33.0	4.0 ± 1.0	sigmoid, 1000, 0.001
Antihistamines	0.05 ± 0.01	0.85 ± 0.19	0.2 ± 0.19	0.01 ± 0.07	0.01 ± 0.05	136.0 ± 22.0	5.0 ± 1.0	sigmoid, 1000
AtypicalAntipsychotics	0.16 ± 0.01	0.95 ± 0.04	0.34 ± 0.07	0.18 ± 0.05	0.17 ± 0.03	459.0 ± 18.0	26.0 ± 2.0	sigmoid, 10000, 0.001
BetaBlockers	0.05 ± 0.01	0.83 ± 0.05	0.65 ± 0.07	0.47 ± 0.05	0.14 ± 0.02	708.0 ± 83.0	10.0 ± 1.0	sigmoid, 1000
Calcium...Blockers	0.13 ± 0.01	0.93 ± 0.03	0.48 ± 0.04	0.34 ± 0.04	0.21 ± 0.03	484.0 ± 19.0	21.0 ± 2.0	sigmoid, 10000, 0.001
Estrogens	0.29 ± 0.02	0.96 ± 0.02	0.48 ± 0.04	0.25 ± 0.04	0.29 ± 0.03	132.0 ± 3.0	13.0 ± 1.0	sigmoid, 10000, 0.001
NSAIDS	0.17 ± 0.01	0.99 ± 0.02	0.48 ± 0.04	0.37 ± 0.03	0.26 ± 0.02	153.0 ± 9.0	7.0 ± 1.0	rbf, 1000, 0.001
Opiods	0.02 ± 0.0	0.99 ± 0.04	0.63 ± 0.03	0.61 ± 0.04	0.11 ± 0.01	726.0 ± 42.0	5.0 ± 1.0	rbf, 100, 0.001
OralHypoglycemics	0.28 ± 0.01	0.95 ± 0.03	0.33 ± 0.04	0.04 ± 0.03	0.06 ± 0.08	179.0 ± 3.0	32.0 ± 2.0	rbf, 1000, 0.1
Proton...Inhibitors	0.05 ± 0.0	0.92 ± 0.08	0.37 ± 0.08	0.27 ± 0.03	0.11 ± 0.01	576.0 ± 44.0	11.0 ± 2.0	sigmoid, 100, 0.01
Skeletal...Relaxants	0.01 ± 0.0	0.86 ± 0.2	0.16 ± 0.16	0.01 ± 0.06	0.01 ± 0.01	806.0 ± 28.0	4.0 ± 0.0	sigmoid, 100, 0.001
Statins	0.03 ± 0.0	0.98 ± 0.03	0.22 ± 0.07	0.18 ± 0.05	0.07 ± 0.01	1655.0 ± 26.0	15.0 ± 1.0	rbf, 100, 0.001
Triptans	0.04 ± 0.0	1.0 ± 0.0	0.07 ± 0.05	0.03 ± 0.04	0.02 ± 0.03	324.0 ± 0.0	5.0 ± 1.0	sigmoid, 100, 0.001
UrinaryIncontinence	0.17 ± 0.01	0.95 ± 0.09	0.44 ± 0.06	0.28 ± 0.05	0.23 ± 0.03	122.0 ± 13.0	7.0 ± 1.0	sigmoid, 10000, 0.001

[7]Parameter — kernel, C, gamma

6.5 Threats to study validity

This study is limited by threats imposed on its external validity as highlighted for the complexity study in Section 5.7. In addition, notwithstanding the fact that the datasets used in this study cut across two fields - SE and healthcare, there is still not enough evidence to generalise the findings. Only four reviews have been used from SE and three of them addressed similar topics while the medical review datasets are relatively small in size.

The study is also affected by a conclusion validity from the indication that including the bibliography data may improve model performance. Further investigation is required to explain or establish the noted differences across the datasets. The

performances observed in this study are limited to SVM models and the feature representation types used. Though SVM has been reported as one of the leading text classification models and often used in automatic CS research.

6.6 Discussion

The overall results of the study indicate that the effect of adding the bibliography data to the input data is uncertain. This may be due to the low reference retrieval rate (generally below 50%) recorded in the DERP datasets which are in the majority.

If the SE datasets where the average reference retrieval rate are approximately 95% were considered in isolation, it can be seen from Tables 6.4a and 6.4b that there is an improvement in the recall with the TiAbs(MeSH)Ref data in three of the four datasets. In Table 6.5, the TiAbs(MeSH)Ref data show equal or higher recall in three reviews. This pattern is repeated with the non-linear kernel in Table 6.6. In Table 6.4, the TiAbs(MeSH)Ref data (Table 6.4b) shows higher WSS performance in two reviews, equal performance in one and lower in one. However, in the Word2vec representation, the TiAbs(MeSH)Ref data (Tables 6.5b and 6.6b) record higher WSS values in three of the four reviews. On closer inspection, the dataset where the WSS values were less in these two cases is the Kitchenham review which is the smallest among the datasets. This pattern was also noticed with the MCC where the TiAbs(MeSH)Ref data had higher MCC values in three of the four SE datasets except the Kitchenham (see Table 6.5 and Table 6.6). The WSS performance of the TiAbs(MeSH)Ref data was however the other way around for the binary representation where the TiAbs(MeSH) data had higher values in all four datasets.

In terms of complexity, it was observed that the three relatively large SE datasets used on average, only about 30% of their training data as support vectors against an average of about 90% in other smaller sized datasets. This showed that the models from these (larger) datasets had likely learned to generalise better, and are likely to be more robust and less complex than those from the smaller datasets. This further emphasized the importance of data volume in the learning of the ML algorithms during the training phase. In SVM, the smaller the ratio of the support vectors used, the better the model had learnt from the data pattern and thus, the better it can generalise over other examples.

Based on the findings of this study, it is not clear yet whether adding the reference information to the datasets can automatically increase the performance of a TM model for automatic CS. More work needs to be put into investigating the factors that contributed to the improved performance in some cases and not in others. Nevertheless, this study has shown that the chances of sustaining or recording an improvement in a model's performance by adding the bibliography information is higher than the chances of recording a lower one.

We note in the retrieved bibliography data that (usually) only one of the authors' names is fully spelled out. This may result in loss of information that may be vital to the establishment of an association between articles that might have cited similar authors due to a common subject since the initials are removed during preprocessing leaving only one name each from the authors. Access to the full author names in the databases could have aided more, the discrimination of the documents. The same situation affects abbreviation of journal names which could have contributed to linking articles with similar journal names.

6.7 Summary

The study reported in this chapter has investigated the impact of adding the bibliography data to title and abstract for the purpose of building TM models for automatic screening of citations in SRs. 19 review datasets were used in the study, four from the SE domain and 15 from the medical domain. Two different sets of the data were prepared from each of the datasets, one with titles and abstracts (with MeSH for the medical data) and the second with bibliographies added to the titles and abstracts. These were used to build and compare SVM models with binary and Word2vec features as was reported in Section 5.4.

The results of the study have shown that the TiAbs(MeSH)Ref set exhibited higher or equal recall, MCC and WSS in the three larger SE datasets with the different feature representations and model kernels. The performance varied when it comes to the smaller DERP datasets and one SE dataset. However, there were more instances of higher or equal performances than lower. No distinct pattern could be established for the complexity in the smaller datasets, but the three larger SE datasets used fewer number of support vectors across when augmented with the reference information. Given the pattern established in this study, no definitive conclusion could be drawn on the impact of the bibliography information on the performance of the CS models. It was however clear that the inclusion of this data is more likely to improve or sustain the model performance than impair it.

CHAPTER 7

TeMACS - A CS Tool

The review presented in Chapter 3 identified a lack of some essential information in CS studies using TM techniques. The review also identified five tools that has evolved from the studies. A lack of reproduction was identified among the studies. A work undertaken to investigate the reproducibility of CS studies is presented in Chapter 4. The reproducibility study identified a set of information to aid the reproducibility of TM based CS studies. The need to report complexity related information was found in the complexity assessment study reported in Chapter 5. In this chapter, a TM tool for Automatic CS (TeMACS) is presented. The key features of how the tool supports transparency by the way of conforming with the findings of previous studies in this research are presented. More design and development details are presented in Appendix D.

The tool combines the findings and recommendations from the various work highlighted above to show how a tool can provide the type information necessary for transparent reporting when used to conduct CS research. The tool provides feedback about its operations to aid the reproducibility and technical understanding of the model used. *TeMACS* is a document classification tool particularly useful for a repeat SR where labelled data from a previous review can be made available to train a model and use it to classify the data for the new SR. It can also be used to classify large datasets where a part of a dataset can be labelled to train a model in order to classify the rest. Once a model is trained, users can then use it multiple times to re-classify data of future reviews on the same subject. The motivation behind the development of this tool, its main features and possible effect on the provision of support for CS in SRs and the research community is presented.

7.1 Introduction

As discussed in Section 1.1.5, a number of studies have been undertaken to investigate tool support for automatic screening of citations in SRs. A collection of studies

across healthcare and SE reporting the potentials of TM techniques for the automation of the CS stage in SRs were identified in (O'Mara-Eves et al., 2015). A review focussed on the quality of information provided in these studies which narrowed the 44 studies reviewed in (O'Mara-Eves et al., 2015) to 35 and identified additional nine primary studies was conducted (see Chapter 3). A variety of methods being proposed were found and five of them have been packaged as a tool. Whilst these studies and tools are useful, reproduction and independent validation of their results and processes still remains a challenge. This is a situation that may affect the timely evolution of a sustainable solution to the problem of automatic CS using TM based techniques borne out of complementary and collaborative efforts, and independently reproduced results.

There is an ongoing effort to increase the awareness of available TM tools to support SRs as a whole or any of its stages. For example, the AHRQ published a white paper which identified 111 TM based tools that partially or fully support the conduct of SRs (Paynter et al., 2016). The list is comprehensive and considers support of SR conducts in general (including commercial tools). Thus, most of the tools reported are out of the scope of this project. However, three relevant tools that are part of the studies reviewed in Chapter 3 listed are:

i) Abstrackr - a web-based tool using active learning for document classification (Wallace, Small, Brodley, Lau, & Trikalinos, 2012). This tool has been independently evaluated and reported in (Rathbone et al., 2015).

ii) Gapscreener - a free SVM-based stand-alone application for automatic CS (W. Yu et al., 2008).

iii) Rayyan - a SR tool with an integrated TM technique for CS (Khabsa et al., 2016).

While SWIFT-Review (Howard et al., 2016) is a tool from a relatively new study, Fastread (Z. Yu et al., 2016) is a tool from a study still being reviewed and are thus not captured on the list. The techniques in both tools have been discussed in Section 3.5.

This chapter introduces *TeMACS*, a TM based tool for automatic CS developed for use by both reviewers and CS support tool researchers across any discipline (see Figure 7.1). It is a simple document classification tool useful for the purpose of automatic screening of citations in situations where previous labelled data of the same subject could be made available for the purpose of training an initial model. The classification and reporting approach in the tool had been heavily influenced by the work reported in this thesis (e.g. the feature representation used, the classifier and outcome data reported to users). *TeMACS* was developed using a Model-View-

[0]the tool can be accessed through http://bitly.com/temacstool

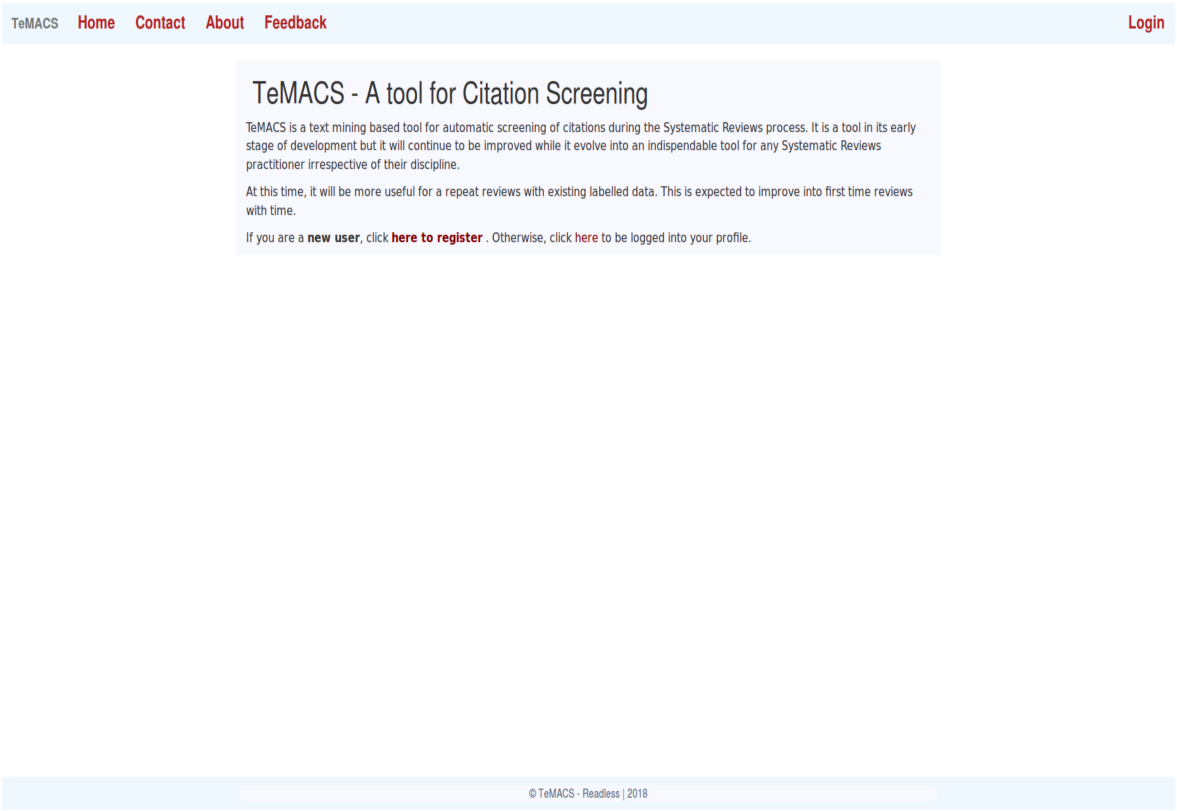

Figure 7.1: TeMACS home screen

Controller (MVC) paradigm with Python 2.7, FLASK, redis, redis-queue, MySQL and jQuery.

7.2 *TeMACS* features

The main features of *TeMACS* are described in this section; namely, creating a 'project', creating a 'new model', reusing 'existing model' for new prediction, 'load data', 'view data' and 'build model'. The high level depiction of transition of the 'views' is shown in Figure 7.2. The tool's key operations are further discussed in the following sections.

7.2.1 Create project

A system user (reviewer) can create a new project by providing a name for the project (Figure 7.3). The process creates a project with the 'project_name' if it does not already exist and updates the 'projects' field of the reviewers table. A new model is also automatically initiated as part of the process.

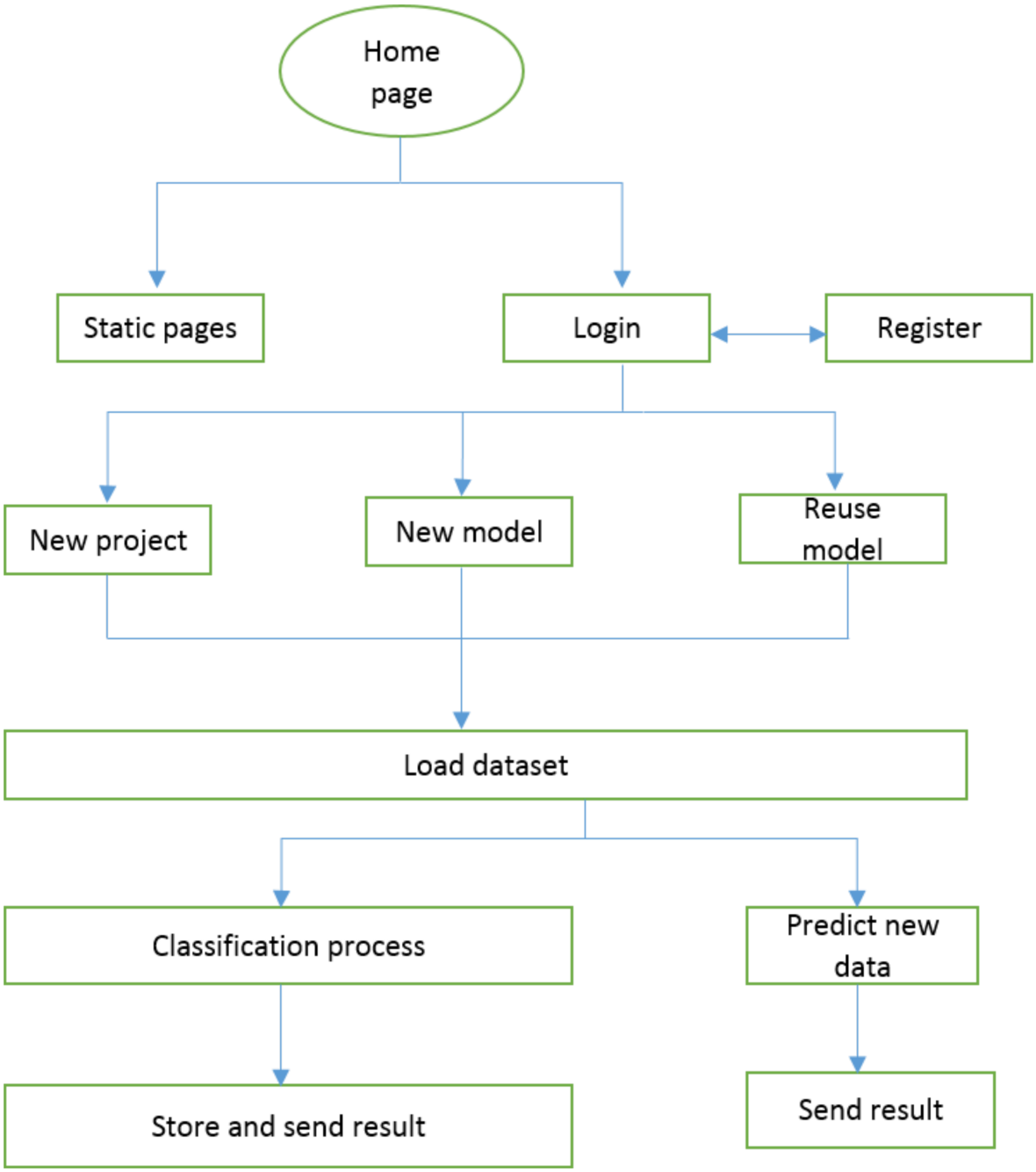

Figure 7.2: High level information flow in *TeMACS*

7.2.2 Create new model

A model is automatically created as part of a new project creation. However, users are also able to initiate the creation by providing the name of the parent project for the model they intend to initiate (see Figure 7.4). The model creation process queries the 'name' field of the 'models' table of the database to ensure the model's name is unique. During model creation, the 'models' field of the 'projects' table is updated.

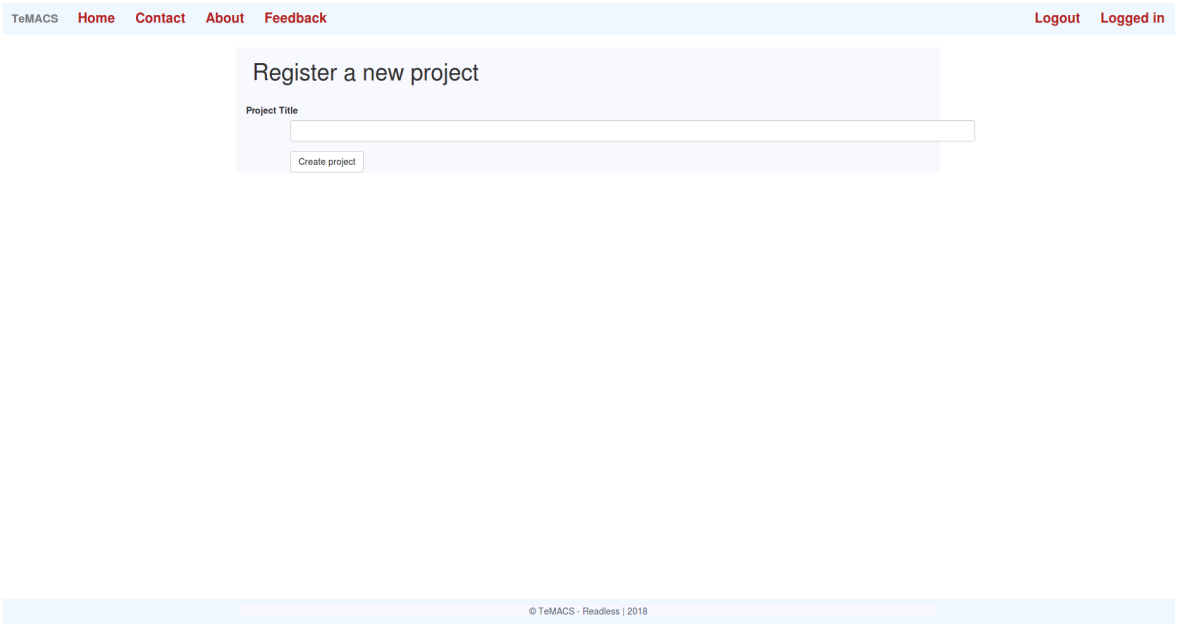

Figure 7.3: New project creation screen shot

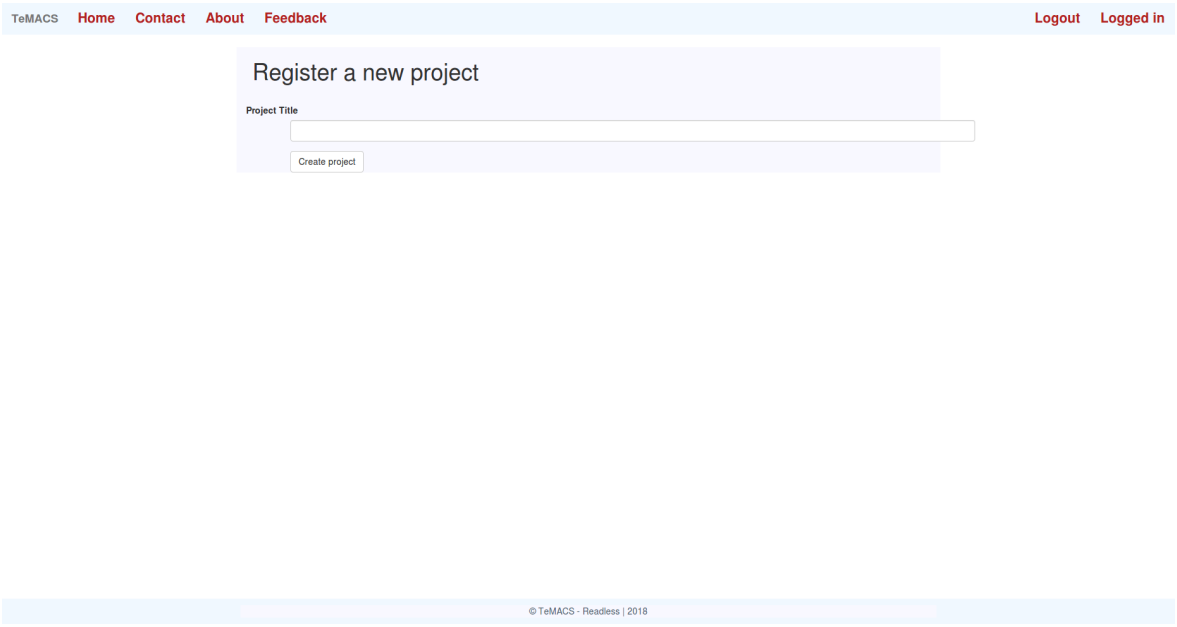

Figure 7.4: Screen shot of the new model creation page

7.2.3 Load data

After users have successfully submitted the request for a 'new project' or 'new model' or 'reuse model' they can then upload their dataset by navigating to its location on their system and selecting the file which must be in a 'comma separated values (csv)' format (Figure 7.5). Users must click the 'upload' button to upload and view top ten rows of the uploaded file and a distribution of the relevant and irrelevant articles in the corpus (Figure 7.6). If the wrong data has been chosen, users have the choice to go back and chose a different dataset. If satisfied with the dataset, users can initiate the TM process by clicking the 'build model' button (Figure 7.6).

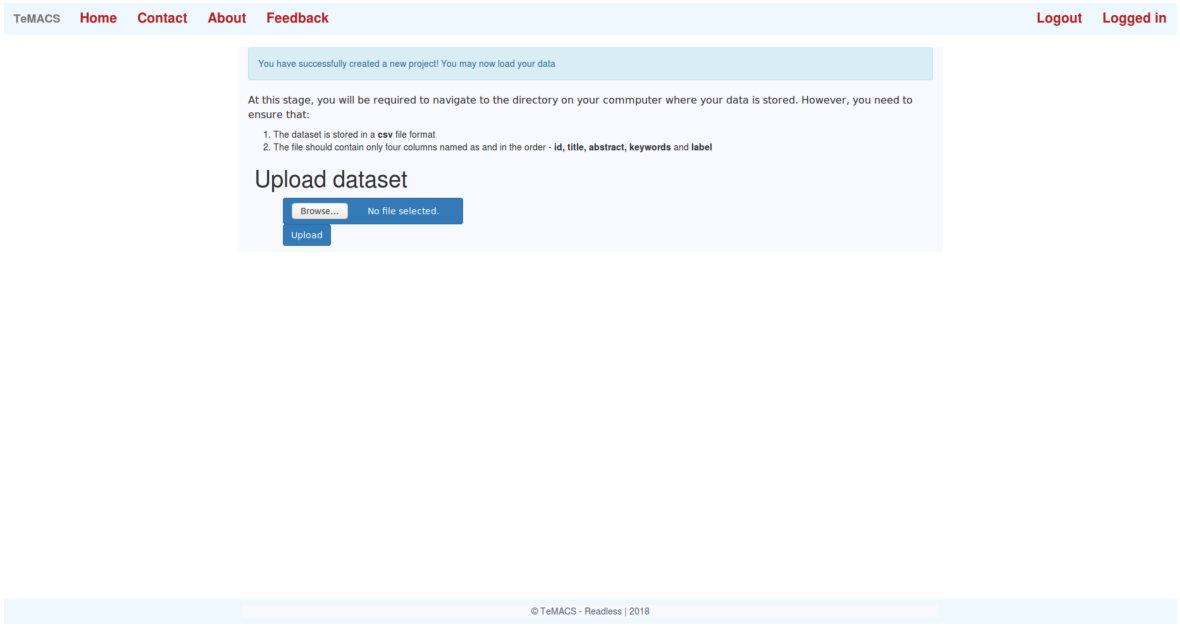

Figure 7.5: Screen shot of the load data page

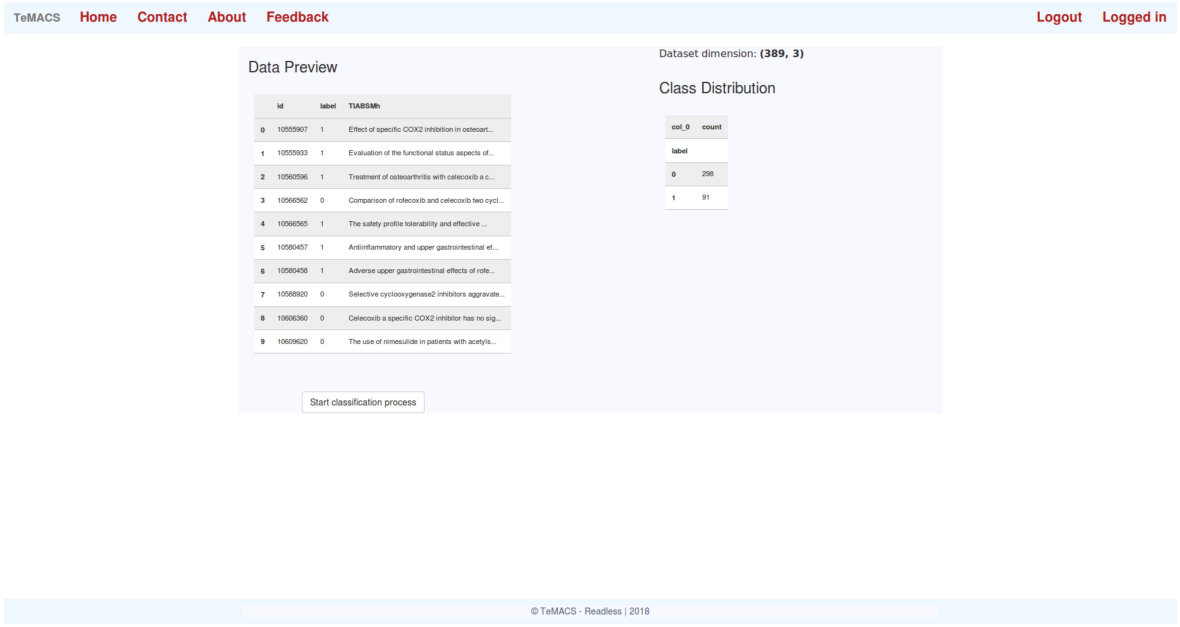

Figure 7.6: Screen shot of the view data page

7.2.4 Build model

Users can build classifiers based on their dataset by clicking the 'build model' button on the 'view data' view (Figure 7.6). From this point on, the whole document classification process takes place in the background. When the background task is initiated, a 'task ID' is generated and the 'job_id' field of the 'models' table is updated; the background task is polled at intervals to determine its state of execution, the view is updated intermittently when the process is ongoing (Figure 7.7).

The architecture for the process of running the classification process in the background, polling for update and updating the view is presented in Appendix D.1.2

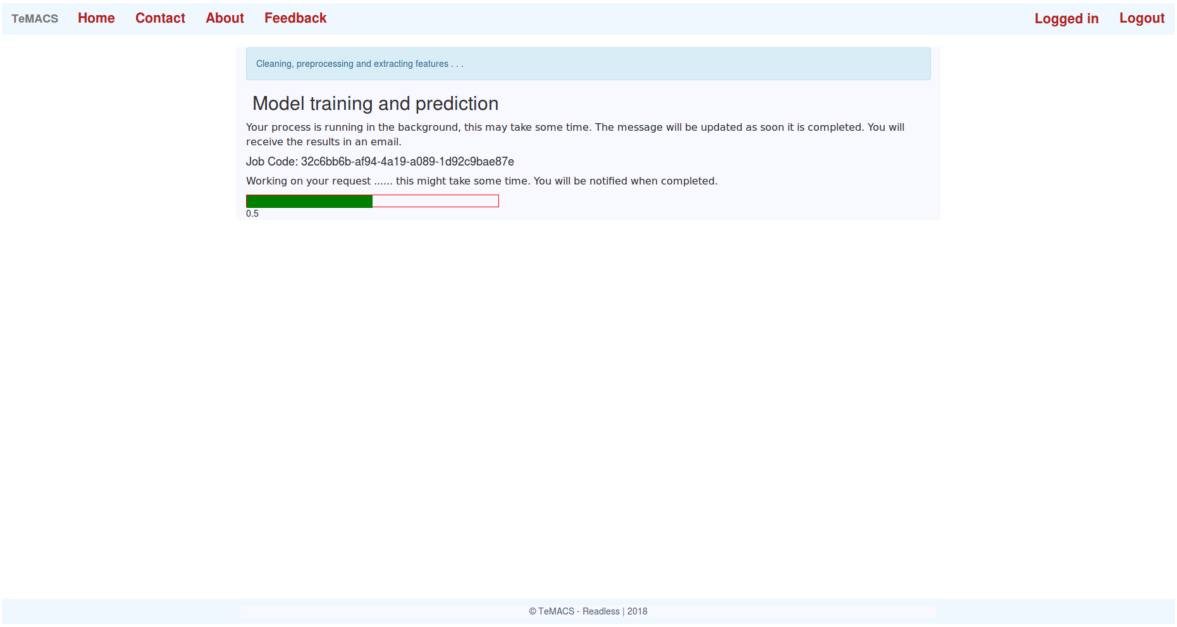

Figure 7.7: Screen shot of ongoing classification process

(Figure D.7).

When the process is completed, multiple fields of the 'models' table are updated. The following information are saved to the database:

i) seeds - a 'dict' structure of the different seed values used in the classification process. They are:

   a) shuffle_seed - used for the initial shuffling of the dataset before selecting half of it for the purpose of selecting the best fitting parameters.
   b) split_seed - the seed value used in the splitting of the dataset initially for selecting best parameters.
   c) gridsearch_seed: the seed value used during the grid search process to select the best parameters.
   d) cv_folds_seed - a list of seeds used in the 'stratifiedKfold' module for partitioning the dataset during CV.

ii) best_model_params: a 'dict' structure of the best model parameters for each model and feature representation types.

iii) feature_vec - a string pointing to the directory of the location of fitted (trained) feature representation object for future transformation of new data.

iv) chi_object - the file directory of the trained χ^2 object for future transformation of new dataset.

v) trained_models - the file directory of the trained classifier objects that can be used for future prediction without retraining.

vi) assessment - a record of the models' performance during development. Only the four primary metrics - TP, TN, FP and FN are recorded.

vii) included - indices of the dataset predicted as positive by the model.

viii) excluded - indices of the dataset predicted as negative by the model.

ix) start_time - a record of the classification process' start time.

x) end_time - a record of when the classification process terminated.

xi) software_env - a record of the names and versions of the software packages used in the process.

The view is also updated to indicate completion as shown in Figure 7.8. An email of the results is subsequently sent to the registered email of the user. A sample of the email is shown in Figure 7.9.

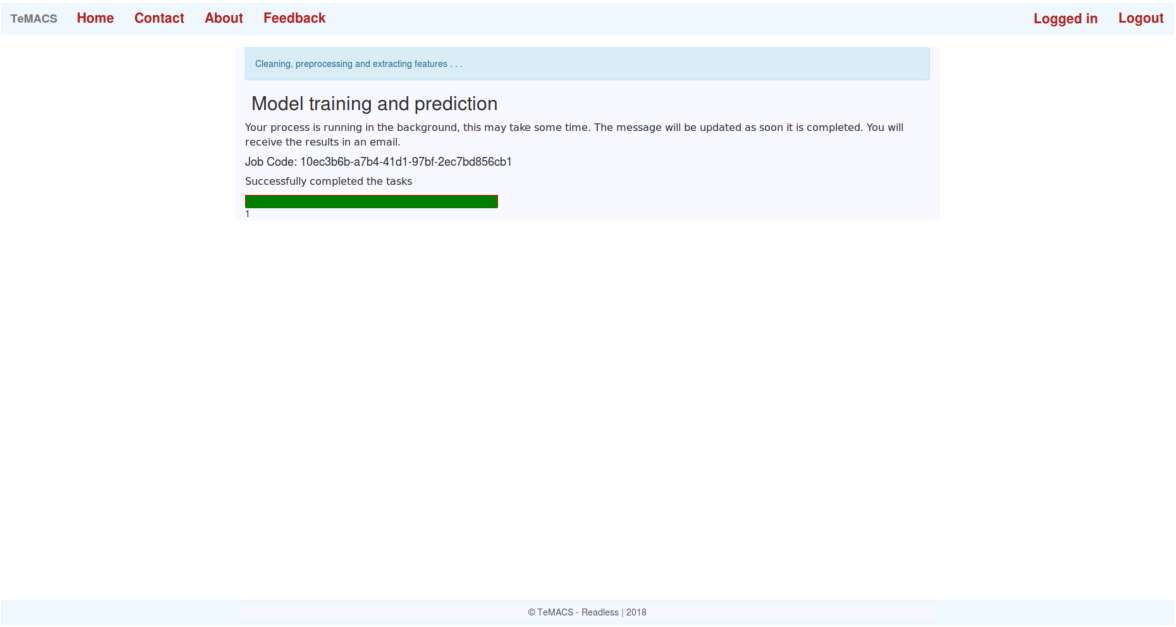

Figure 7.8: Screen shot of complete classification process

The background process follows the TM steps discussed in Section 2.2 and depicted in Figure 2.4. The particular activities of each step as it pertains to the application are described below.

7.2.4.1 Data retrieval

The data in this case is provided by the user and uploaded through the 'load data' view (Figure 7.5). The dataset is stored temporarily in a folder during the classification process and deleted after its completed. The raw user data is not stored permanently by the application, only derived artefacts like the trained feature vector and classifier are stored for the purpose of reuse.

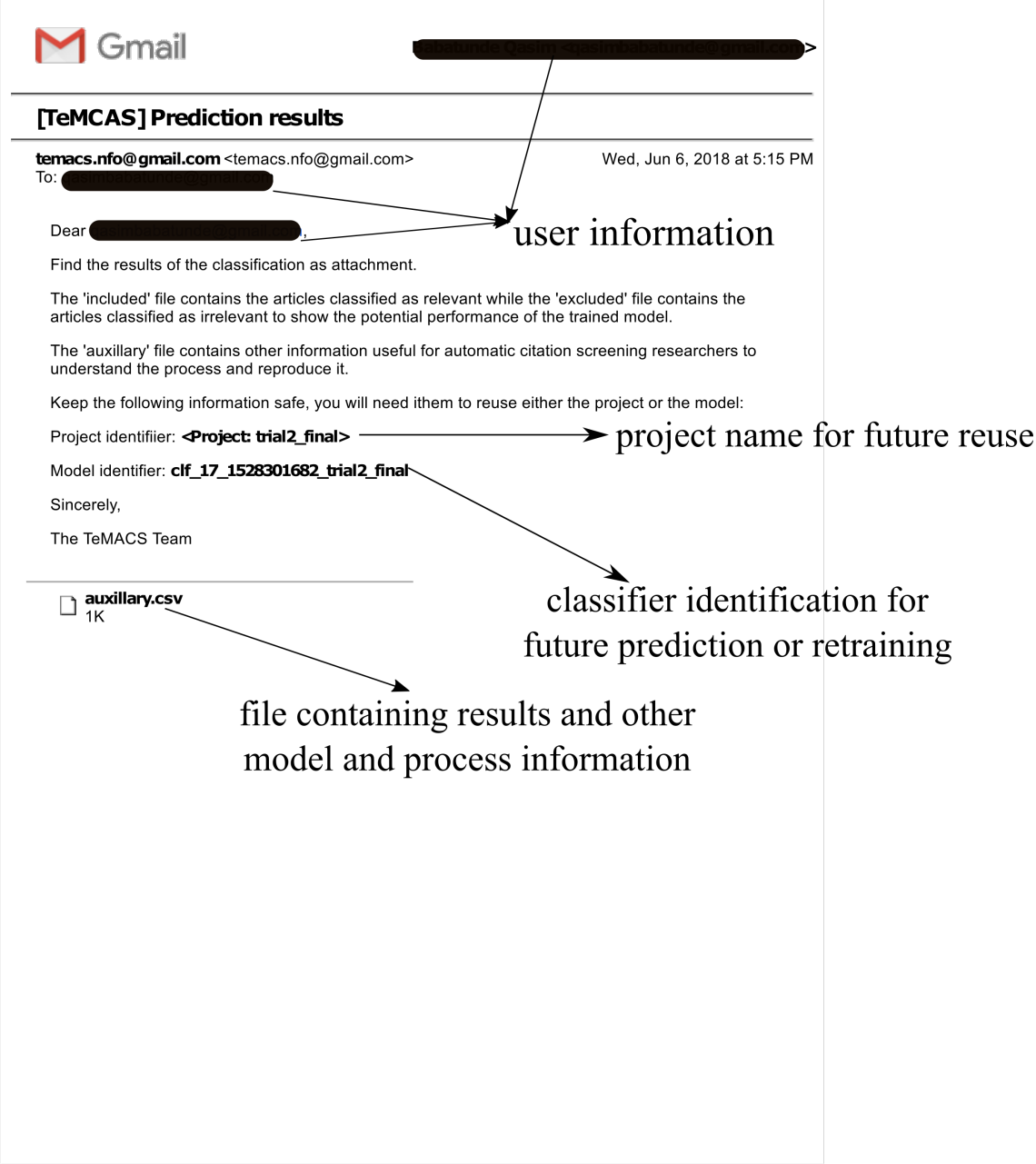

Figure 7.9: Screen shot of the email sent on completion of the model training and prediction

7.2.4.2 Parameter selection

The second step during the course of 'build model' in the application is to select parameters (C and gamma) for the SVM classifiers. This involves three activities:

i) Data shuffle: Prior to any action, the data is initially shuffled with a captured seed value.

ii) Data split: After shuffling, the dataset 50% of the dataset is used to train tem-

porary models to determine the parameters to use in training the final model. This is done for each feature representation type - binary and Word2vec.

iii) Initial fit: The 50% of the data from step 'ii' is used in this step to fit primary models to determine parameters for use in the final classification. The process of selecting the parameters of the 'best model' also involves determining the optimal α-value to use in the χ^2 method for feature selection.

The 'best model' for each feature representation is 'pickled' and saved on the hard drive while their storage locations are saved in the database.

7.2.4.3 Preprocessing

The features were transformed separately into binary and Word2vec representations. Other preprocessing included the steps described in Section 2.2.2 except stemming.

7.2.4.4 Dimensionality reduction

The χ^2 statistic (introduced in Section 2.2.4) is used for feature selection. The α-value to select the top features is automatically determined using the 50% data split as described during 'parameter selection'.

7.2.4.5 Model training

The three types of models based on the two feature representation types that had been used in this research are trained in this tool. Models are trained with the saved best models on the whole dataset using 5×5-fold CV. The training prediction for each fold is stored, the predictions from each model at the end of the CV process is ensembled and a final prediction selected through 'voting'. Only the basic metrics of TN, FN, FP, andTP are reported. Every other metrics can be further independently calculated from these metrics as desired by any user. These predictions and metrics are sent to the user at the end of the process.

7.2.4.6 Final models

At the end of the model building process, a final feature vector, χ^2, and classifier objects are fitted over the whole dataset and saved for future reuse. These are used to process dataset when new predictions with existing models are initiated. Figure 7.10 shows the snapshot of a folder containing the final trained artefacts for a particular dataset. The artefacts combined with the information provided to the user as mentioned in Section 7.2.4 makes model re-use and re-training possible.

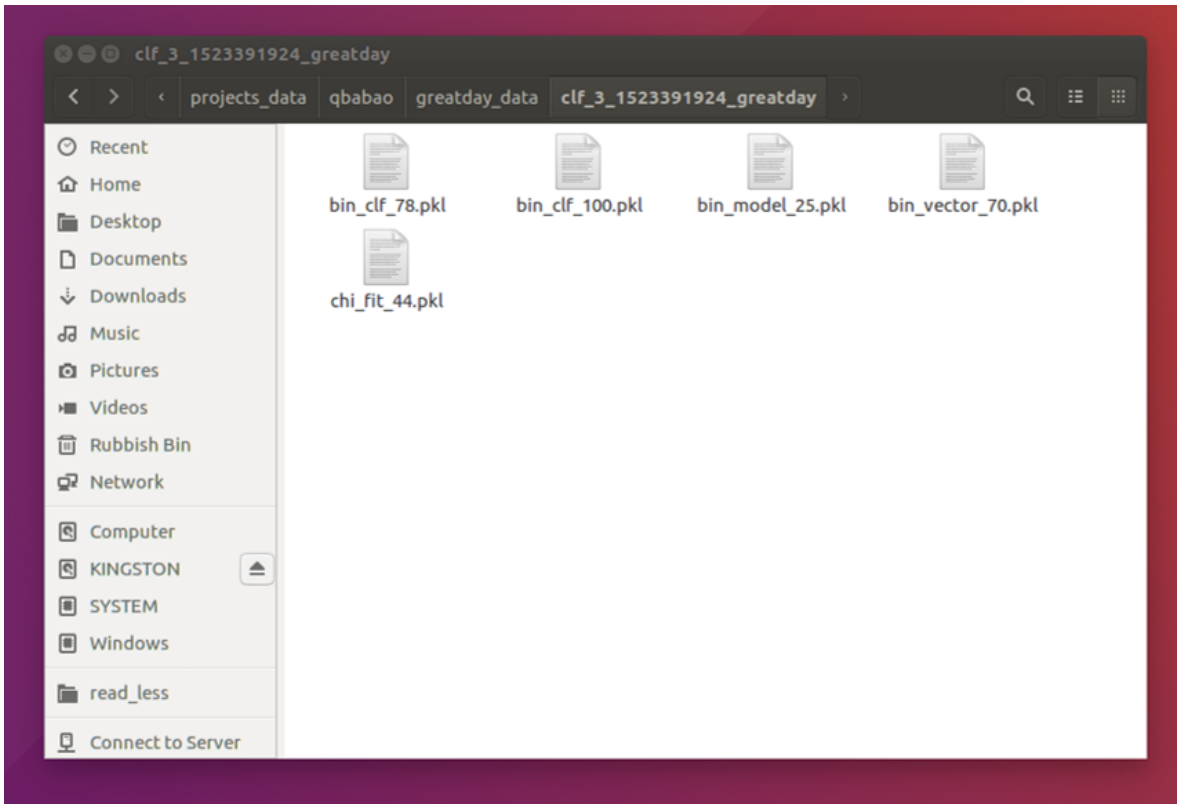

Figure 7.10: Screen shot of folder containing the trained classification model and feature vectors saved for future reuse

7.2.5 Reuse model

There are situations where reviewers(also called users) will want to update a previous review, in such cases a previously trained model in *TeMACS* on a the same topic can be re-used to automatically screen the new citations. Users are able to reuse an existing model for new predictions. The user will need to provide the name of the parent project and the specific identifier of the model to use (Figure 7.11). This will query the 'name' fields of the 'projects' and 'models' tables to establish existence and relationship between the model, project and the user. The predictions will be emailed to the user's email after the process execution. The information necessary to reuse a model would have been made available to the user during the initial training of a model on the subject (see Sections 7.2.4 and 7.2.4.6).

7.3 *TeMACS* reproducibility support

The development of *TeMACS* is driven by the finding of the various studies in this research. Particularly, the need for more transparency to enable reproducibility and increase model quality understanding through provision of complexity details. *TeMACS* is developed as a prototype to demonstrate how a tool for CS can be transparent to ensure its processes are reproducible and understandable. In order to ensure this,

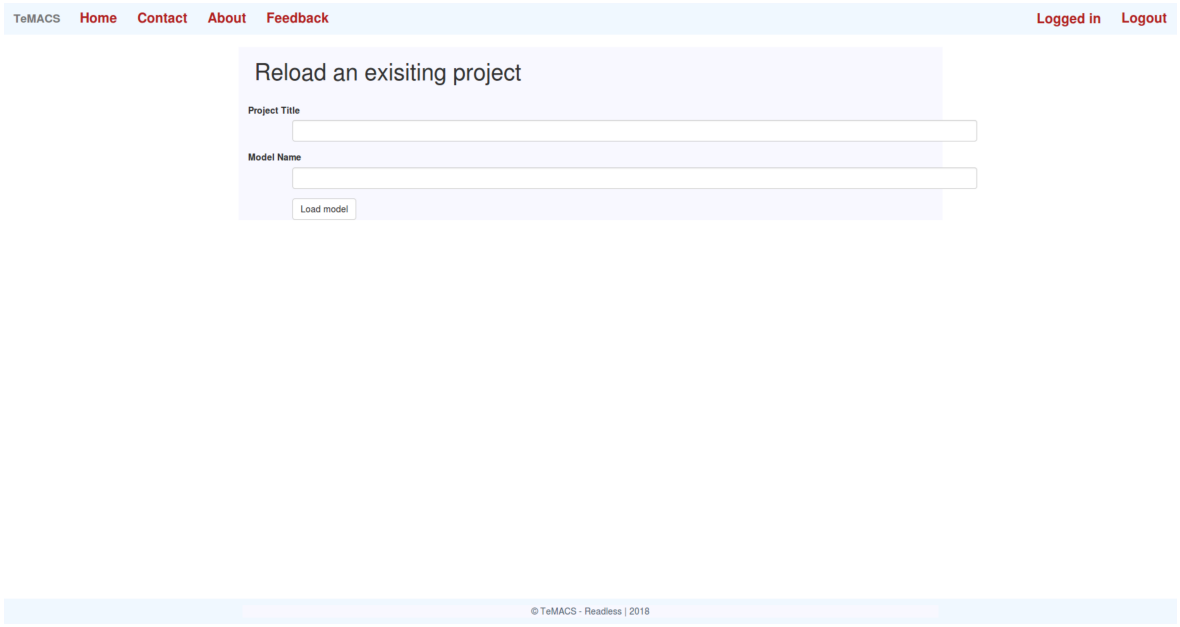

Figure 7.11: Screen shot of the model reuse page

during the model training stage as described above (refer to Section 7.2.4), the tool saves data useful for the reproduction of its process driven by the propositions in the revised reproducibility checklist presented in Section 4.5.2 and updated in Section 8.3.1. The information stored by the tool and made accessible to users are listed in Section 7.2.4).

The tool does not retain the original dataset. Instead the trained objects generated from the data are stored to facilitate reusability. It also provides various seed values to recapture the state of randomised processes (data split, shuffling and modelling) used during each fold of the CV. The right over the data download link (as recommended in the checklist, Section 8.3.1) still resides with the user, as the tool does not store any raw data. In place of feature representation, final feature vector links, the tool stores the trained models for these artefacts including the trained classification models. The parameters of the different SVM algorithms are also provided. It is believed that these set of information as informed from the checklist will support the reproducibility of the tool's process.

7.4 Limitations of the *TeMACS*

A key limitation of *teMACS* is the supervised learning algorithm it implements since a labelled set of data which is not usually available is required to train a model. Effort in future updates will be geared towards experimenting with and incorporating a semi-supervised approach.

Another limitation is its support for single users only. A typical review involves multiple reviewers, the current version of *teMACS* supports only single user mode.

A future update will incorporate the multi-user collaboration mode. The tool had been developed as a prototype. Its functionality as described in this chapter has not been independently verified neither has the its support for reproducibility being independently verified as well. As a new tool, it takes some time to start receiving user feedback. It is anticipated that the feedback will support the presentations in this chapter as care has been taken to ensure its functionality.

7.5 Conclusions and future direction

This chapter has introduced *TeMACS*, a TM driven tool built to support reviewers in automatic CS during the conduct of SRs. It also aims to support TM based automatic CS research by providing information to improve the reproducibility of its process and technical details of its models that may be used to determine, for example, their complexities. This tool was developed in response to the lack of a tool whose operations are made transparent enough so that other researchers can evaluate, understand, reproduce and be able to extend it (possibly through collaboration). In line with the objectives of this project, the implementation has shown that it is possible to have an independent tool to conduct a citation screening research and have enough details of the operation to prepare the report of the exercise besides the classification results.

A major part of the methods implemented in *TeMACS* was informed by the findings of studies reported in this thesis. For example, the reproducibility study (Chapter 4) informed the type of details captured besides the classification results. The feature representations that were explored with the SVM algorithm (in Chapters 4, 5 and 6) were implemented in the tool alongside ensemble method reported in Chapter 3 to be the second most used method beside the SVM.

The tool will be made available free for public use with codes accessible from 'Github' for interested researchers to contribute to its development and continuous evolution. Users can also send feedback and suggestions through the application interface.

On future development for *TeMACS*, the methods implemented in the tool will be increased with more flexibility for user options on their preferred methods and classification approach. The extensions will also include integrating automatic data retrieval from a number of databases. Algorithms will be optimized and updated as need be, existing implementations will also continue to be refined for improved user experience. A foreseen risk is the possibility of increasing difficulty in managing the extensions, software updates and dependency between different packages as the system grows. Therefore, the lifespan and relevance of *TeMACS* will rely more on the involvement, contribution and support from the SR and automatic CS communities. Efforts at publicity will be geared towards its integration into the SR toolbox (Mar-

shall & Brereton, 2015) and it will be presented and demonstrated among the SR and automatic CS research communities.

CHAPTER 8

Discussion

Chapter 7 describes the features of a 'transparent' CS support tool and the motivation behind it. The outcome indicates the possibility of a tool useful for supporting CS in SRs and at the same time be transparent enough to promote research on the subject through operational transparency. This chapter aggregates all the work undertaken and reported in this thesis and qualitatively discusses them against the original research questions listed in Chapter 1.

8.1 Introduction

The main aim of this thesis was to investigate experimental transparency in studies using TM techniques to support CS in SRs; vis-á-vis the extent of technical information reported and how the information affects reproducibility of the studies and understanding of the complexity of the models. As part of this investigation, a 'transparent' tool to support CS in SRs and its research was developed.

Three research questions were listed in Chapter 1 to be answered in order to fulfil the set aim of this research. The questions once again are:

> **RQ1**: What information is required to improve experimental transparency in studies reporting the use of TM techniques for automatic CS in SRs?
>
> **RQ2**: What information is essential to the reproducibility of TM for CS studies?
>
> **RQ3**: What information about model complexity should be included in TM based CS studies?

In addressing RQ1, the need for more transparency in TM techniques based studies to support CS, premised on the findings from the work undertaken to investigate this, is presented in Section 8.2. A response to RQ1 based on the findings from the work is presented in Section 8.2.3. The work undertaken to provide a response to RQ2, the level of reproducibility of current studies and identified information to aid

study reproducibility are established and discussed in Section 8.3 with a focussed response to RQ2 presented in Section 8.3.2. In response to RQ3, the work undertaken to ascertain the possibility of current models being complex and motivate the need for focussing on it starting from reporting is presented in Section 8.4. A response to RQ3 is presented in Section 8.4.3. A discussion on the CS tool developed with combined findings from the different work conducted in this project is presented in Section 8.5. Future directions and recommendations for CS tool researchers and users are highlighted in Chapter 9.

8.2 Experimental transparency in CS studies

In this section, a discussion of the work undertaken to explore the current TM techniques and models to support the CS stage in SRs is presented. Research activities, undertaken to investigate the level of transparency in TM-based studies to support CS (or study section as referred in some domains) include:

i) **Literature review** to identify TM methods for CS support and information reported about them.

ii) **Reproducibility assessment** to evaluate the reproducibility of CS studies and identify the effect of relevant factors.

The overall findings from these research activities are brought together in Section 8.2.3 to provide a summarized response to RQ1.

8.2.1 Literature review

In Chapter 3, a literature review to identify TM methods and practices being explored for automatic CS in SRs was reported. A variety of classification methods (supervised learning) were found being mostly explored. A small number of studies also explored clustering (unsupervised learning) with visualizations. The results revealed a young but growing field with promising results. The SVM and the ensemble methods have attracted the most research attention with the trend recently shifting towards semi-supervised learning approach, particularly, the active learning method. Owing to the nascent stage of the field, only five of the 44 studies reviewed have resulted in a tool for general use. The focus of the studies has been more on saving reviewers' time and effort during CS. This is even more evident in the focus on recall, precision and one of the custom metrics proposed - WSS - which has been found used in several other studies as a measure of the methods' potential. The amount of time and effort consumed by SRs activities is one of the concerns raised by the SRs community as discussed in Chapter 1 (see Section 1.1.4). Results reported in the studies have not been independently reproduced; only two of the tools have received independent

evaluation. The studies are however positive about the potential of their results and the independent evaluation of the tools have been positive. Based on the results of the review, there is scope to perform a transparency investigation of the studies to establish the sufficiency of the information made available. Particularly as it concerns reproducibility and understanding of the models' complexity - two key areas an emerging computation research field. The literature review made the following contributions to the project:

i) Current TM methods for CS and their usage were identified.

ii) Available TM based tools (evolving from the reviewed studies) were identified.

iii) A need for more transparency in the studies was established.

8.2.2 Reproducibility assessment

Subsequent to the literature review, a reproducibility assessment study was conducted to investigate the reproducibility of the CS studies and identify factors that contribute towards it. The investigation took the form of a qualitative assessment. It consisted of three main activities: an initial attempt to actually reproduce six of the studies that are based on common datasets, identification of the factors that contributed to the reproduction attempt and the development of a systematic assessment framework, and a systematic assessment of 33 studies based on the framework. The framework was inspired by the work of González-Barahona and Robles (2012). The studies were strictly assessed based on the information provided in the reports. The results of the reproducibility assessment identified elements of TM experiments that are critical to the reproducibility of study results but are often not found in the reports. Undertaking this investigation contributed the following to the project:

i) The feasibility of reproducing TM-based CS studies was investigated.

ii) The lack of information to enable reproducibility of the studies was reinforced. Specifically, studies on CS support with TM techniques were assessed for reproducibility for the first time.

iii) The specific information vital to reproducibility were identified and a checklist produced.

8.2.3 Response to RQ1

What information is required to improve experimental transparency in studies reporting the use of TM techniques for automatic CS in SRs

The findings of this research have determined that automatic CS studies based on TM techniques can do with more transparency in reporting. The studies are concentrated on reporting superficial details on the conduct of their studies and performance

results. These information are sufficient to understand the studies but not enough to, for example reproduce the results or understand and interpret the the models' performance. More of the reports have been about what was done but not how (and with what) it was done. The community need to put more work into the standardization, quality, type or level of information to be provided in studies. This is particularly important given the nascent stage of the research area, the contribution poof a sustainable tool to the conduct of SRs and the role of SRs in providing evidence-backed practice evolution in a discipline. The reproducibility assessment further reinforce the fact that the information currently being provided are insufficient, at least as far as study reproducibility is concerned. Overall, more work is needed in providing a structural framework and reporting guidelines for automatic CS studies.

8.3 Reproducibility essentials

In this section, a discussion of the work undertaken to investigate the information elements critical to the reproducibility of CS studies based on TM techniques is presented. Chapter 4 presented an investigation to determine the reproducibility of CS studies. As recapped in Section 8.2.2, the research takes the form of a three steps activity. The reproducibility attempt considered as a whole was not successful, varying level of reproduction was accomplished at different stages from the different studies. The experience became useful at identifying the role of each element found reported or otherwise. An evaluation framework was subsequently developed to assess 33 studies. The reproducibility assessment lead to the development of a checklist of identified essential study elements to aid reproducibility. 14 information elements were captured in the checklist to be used as a guide for authors and reviewers. The checklist is validated in Section 8.3.1 against the nine literature update that were not part of the initial assessment from which it was created. The results showed a need for improved reporting before the studies can be reproducible. It particularly identified important elements that are often downplayed in studies.

8.3.1 Checklist validation and update

In this section, the outcome of an activity to validate the the reproducibility checklist (see Section 4.5.1) is presented. The validation was conducted using the nine primary studies from the literature review updates (see Section 3.5). This activity was conducted to examine the consistency and relevancy of the checklist's items to new studies and identify any improvement needs. The activity was also conducted to identify any possible change in reporting pattern in new studies. The result of the validation is presented in Table 8.1. Three updates were effected in the new version of the checklist (updated items 1, 4 and 7 are in italics). The justification for adding

Table 8.1: Validation of checklist with nine review update studies

Item No.	Elements	UP1	UP2	UP3	UP4	UP5	UP6	UP7	UP8	UP9
1	*Dataset identification*	Y	Y	Y	Y	Y	Y	Y	Y	Y
2	Original location of the raw dataset	Y	Y	N	Y	Y	Y	Y	Y	Y
	Provided link to local copy of:									
3	a. Raw dataset	N	N	Y	N	Y	N	Y	Y	Y
	b. Target dataset	N	N	N	N	N	N	Y	N	N
4	*Feature set*	Y	Y	N	Y	Y	N	Y	Y	Y
5	Pre-processing details	Y	N	Y	N	Y	Y	Y	Y	Y
6	Feature representation technique	Y	Y	Y	Y	Y	Y	Y	Y	Y
	Dimensionality reduction approach:									
7	a. Feature selection	X	X	Y	X	Y	X	Y	X	X
	b. Feature extraction	X	Y	X	Y	X	Y	X	Y	Y
8	Final feature vector – download link	N	N	N	N	N	N	N	N	N
9	Training algorithm	Y	Y	Y	Y	Y	Y	Y	Y	Y
	Custom algorithm									
	a. Text	X	X	X	X	Y	X	Y	Y	X
10	b. Code	X	X	X	X	N	X	Y	N	X
	c. Algorithm	X	X	X	X	Y	X	Y	N	X
	d. Executable file	X	X	X	X	N	X	Y	Y	X
11	Model assessment method	Y	Y	Y	Y	Y	Y	Y	Y	Y
12	Detailed model assessment result	Y	Y	Y	Y	Y	Y	Y	Y	Y
13	Randomization seed values	N	N	N	N	N	N	N	N	N
	Training/test data partition available or indices provided									
14	a. Link to data partitions provided	N	N	N	N	N	N	N	N	N
	b. (link to) data indices provided	N	N	N	N	N	N	N	N	N
	c. Seed value provided	N	N	N	N	N	N	N	N	N
	Software information									
15	a. Name provided	Y	N	N	N	N	N	Y	N	Y
	b. Version details	N	N	N	N	N	N	N	N	N

the two new items 1 and 4, and amending item 7 is given below:

i) Dataset identification (item 1): During the validation exercise, it became clear that the initial version of the checklist did not provide explicitly for the name of a dataset. This is particularly useful for public and benchmark datasets being utilized for research purpose. The initial item 1 - *Original location of the raw dataset* was found inadequate to cater for this requirement. More so that a location may be a web link and not necessary a name. In a case where the location is missing, the name can assist in locating the dataset.

ii) Feature set (item 4): The feature set refers to the columns of interest from a dataset used for the classification purpose. For example, title, abstract and any other useful columns like keywords, references etc. This item was missing in the version 1.1 of the checklist presented in Section 4.10 but now brought back to replace item 3 in version 1.0 presented in Section 4.8. This is a key element to knowing what features or columns of a dataset were used in certain studies.

iii) Dimensionality reduction approach (item 7): This item is a refinement of items 8 and 9 in Table 4.8 which corresponds to items 5 and 6 in Table 4.10. The amendment became necessary because feature selection and feature extraction are different approaches to reducing the dimensions of the feature vector for improved performance (see Section 2.2.4).

The pattern noticed in Table 8.1 is consistent with the pattern presented in Table 4.10. It is however notable that five (56%) of the nine studies assessed provided a link to the location of their raw datasets against 10% in Table 4.10. An improvement was also found in the preprocessing information provided in updated studies, from 57% to 77% (7 out of nine studies). The same improved trend was noticed in information provided on custom methods, particularly noticeable is UP7 which provided all the four possible details expected about their tool including coding and implementation. The study actually reported being motivated by the reporting and reproducibility challenge identified in the literature review (Chapter 3) presented in (Olorisade et al., 2017a).

8.3.2 Response to RQ2

What information is essential to the reproducibility of TM for CS studies?

With regards to RQ2, the findings of this research has established the fact that the information currently reported in CS studies does not sufficiently support reproduction. Consequently, 14 information elements that are essential to the reproducibility of CS studies based on TM techniques were identified. They include some often overlooked information like seed values not found reported by any study but without which reproducing a study result will be almost impossible. These elements are a

result of the reproducibility investigation conducted on 33 studies. The validation of the elements on nine new articles as presented in Section 8.3.1 showed that the elements are sufficient to evaluate the reproducibility of a study. Further validation and refinement are however required including ranking of the elements in order of importance. The possibility of a tool providing such information as recommended in this study is demonstrated by the output of *TeMACS* as discussed in Chapter 7.

8.4 Complexity reporting motivation

A discussion of the work undertaken to investigate the need to report model complexity related information in CS studies using TM techniques is presented in this section. The research activities, undertaken to investigate complexity concerns in automatic CS studies using TM techniques include:

i) **Complexity assessment** to investigate complexity concerns in CS models to motivate the need to include model complexity information in studies.

ii) **Feature enrichment** to investigate the effect of including the full bibliography data on the performance and by implication, the complexity of CS models.

The findings from these research activities are discussed together in Section 8.4.3 in response to RQ3.

8.4.1 Complexity assessment

The literature review identified an array of general information lacking in the studies for automatic CS. This lack of information was investigated further and narrowed to if and how it affects the reproducibility of the studies in Chapter 4. Information regarding the complexity of the models was also found lacking. Apart from being an important aspect of (statistical) computational models, reporting of model complexity will be fulfilling a basic scientific requirement. The complexity assessment was conducted to investigate the robustness of the models being proposed through their complexity. The study was conducted with the aim that if the investigation finds high complexity in the models, reporting the complexity related information is important. Since it was established from the reproducibility assessment that the studies could not be reproduced, to accomplish the objectives of this investigation, hypothetical models representative of a typical model found in the studies were developed. Multiple feature representations were explored with the SVM algorithm. The SVM makes its classification decision based on the SVs- data nearest to the hyperplanes (see Figure 2.1). Therefore, its complexity is determined by the size of its SVs in relation to its training data size. In addition to other performance metrics recorded, the SV size for each model was also recorded. The results indicated a high complexity across all

the models developed with performances similar to top performances witnessed in the CS studies. The binary feature models with non-linear kernels exhibited relatively higher complexity compared to the Word2vec feature linear kernel models. The contributions of this investigation to the project are as follows:

i) Potential concerns for complex models were found and the need to report complexity related information was established.

ii) Multiple features were explored which revealed differing linearity and complexity in the resulting data and models. It was the first time the Word2vec features were explored in automated CS study.

iii) The first study to explores complexity issues in TM-based automatic CS studies.

The complexity study equally established the need for complexity information to be considered by studies. This will assist independent researchers understand how to particularly interpret future performance of the models as well as improving it.

8.4.2 Feature enrichment

The feature enrichment investigation took the same form and process as the complexity assessment except for the addition of the bibliography data to the feature set. The findings from the investigation were not consistent. Three relatively larger SE datasets showed improvement in performance and lower complexity with bibliography data than without. The performance and complexity varied across the rest of the healthcare datasets and one SE that are relatively small. Nevertheless, the study shows that the addition of bibliography data is more likely to improve a model's performance and complexity than impair them. This investigation contributes the following to the project:

i) It is the first to explore the of use full bibliography data with Word2vec features.

ii) It finds the potential of bibliography data at improving model performance and complexity.

8.4.3 Response to RQ3

What information about model complexity should be included in TM based CS studies?

The findings of this research have determined that complexity information is essential in automatic CS studies. The information will enable proper assessment, understanding and interpretation of the models particularly in the context of future performance. The complexity information will also assist in the reproduction of study results. The complexity indication differ for different learning algorithms, the size of the SVs is used in this research for SVMs. The complexity information relevant to the

algorithm being used should be provided in studies to maintain more transparency on the models' actual performance. In that case, it will be easier for independent researchers to determine if a model has overfit to achieve reported performance or otherwise.

8.5 Transparent CS tool

In Chapter 7, the features of a CS tool - *TeMACS* was presented. The different research work presented in this thesis informed the development of the tool and contributed to its features. The findings from the complexity assessment and feature enrichment studies in Sections 5.4 and 6.3 contributed to the method implemented in the tool. The findings from the literature review in Chapter 3, the reproducibility assessment in Chapter 4 and the complexity assessment in Chapter 5 contributed to the type of information collected and made available to the users when new models are trained. *TeMACS* is the only tool that has been developed with a view to aid other researchers at understanding its processes by providing information which can be used to recreate, reproduce and revalidate its processes and results. Despite being currently useful in situations where labelled data are available or where previous reviews could be recreated to access labelled data, *TeMACS* saves object instances built from its data, therefore, shorter turnaround time for reuse in future review updates and ensuring consistency in the way the data is preprocessed. *TeMACS*' support for reproducibility is hoped to pave way for its timely advancement and a general timely advancement in the research for automatic CS tool using TM techniques. The functionalities of *teMACS* as described in this work remains to be independently verified being a new tool. The extent to which the data it produce also supports independent reproduction of its process will is also yet to be validated by an independent researcher. It is anticipated that other researchers will use the tool and assess its support for reproducibility.

8.6 Threats to research validity

Some of the key threats to the validity of the work presented in this thesis are brought together in this section, categorised under the different types of validity threats. The limitations of *TeMACS* as a tool for automatic CS has bee presented in Section 8.5 with further discussion in Section 9.2.

8.6.1 Construct validity threats

The initial literature reviewed in this project relied on an existing systematic review conducted on similar subject by O'Mara-Eves et al. (2015). Though, the literature list was later updated to cover for the period between when the review was published

and this project, there is a possibility that some paper may not have been covered either in the review by O'Mara-Eves et al. (2015) or the update search. However, every effort is made to ensure that the list covers all available literature until the writing of this thesis by checking continually for any possible new publication. Another approach employed is the checking of the reference list of the new articles identified in the update literature search for any missed publication.

8.6.2 External validity threats

The datasets used in the various work reported in this thesis covers only the healthcare and SE fields. As discussed in Section 1.1.1, SR in other disciplines with their own datasets that are not covered in this research. The healthcare and SE datasets used also covers a small sector of the fields' research themes. The investigations undertaken in this project are however not specific to the data used. The experience can thus be easily generalized within the context of the investigations.

Model complexity is a statistical phenomenon valid for any computational algorithm. Whilst the concept can be generalized across the field, the SVs measured in the relevant work in this research are specific to the SVM algorithm used. Complexity measures of other algorithms are not covered in this research, studies using other algorithms will have to record complexity measures that are suitable to such algorithms.

The model performances of the SVMs reported are not necessarily generalisable. They were generated from datasets that are considered relatively small from the ML perspective and are highly imbalanced. The CV method was nevertheless used to mitigate overfitting. The performances observed are also limited to the parameter settings used and the random (seed) value options. It is impractical to exhaust all possible options particularly of seed values to know which brings out the best performance in the algorithm.

8.6.3 Internal validity threats

The assessment metrics defined in Section 4.2.2 are based on the experience of the researcher guided by two previous researches that had attempted similar definitions. The studies were also assessed based on the researcher's understanding of their contents. It is believed that the reproducibility metrics definition and assessment suffices for the purpose of this work and are subject to future refinement.

8.6.4 Conclusion validity threats

The systematic assessment presented in Section 4.2.3 was conducted subjectively by a researcher. A corroboration from at least one other researcher would have

given more reliability to the outcome. Nevertheless, its not likely the outcome might have changed from having researchers assess the studies because the exercise was conducted purely for research with no preference for any of the studies involved.

There were indications from the feature enrichment study presented in Chapter 6 that adding bibliography data to article features for the purpose of automatic CS could improve the performance of the models. This finding however cannot be definite since the performance was not consistent across all the datasets. This calls for a need for further investigation into the definite effect of bibliographic data addition to citation features or other possible factors that might be responsible for the inconsistency in the models' performance trends. The model performances reported throughout this work are limited to SVM models, the parameters chosen, the feature representation types and the feature ranking methods used.

8.7 Summary

This chapter has combined the findings from all the work undertaken in this project and discussed them against the original aim and research questions set out in Chapter 1. The extent to which the different work undertaken in this research has been able to answer the research questions was clarified. An update to the reproducibility checklist was also presented and validated against the literature review update. The development of *TeMACS* and how it combines the different work of this research was also presented. There still exist opportunities for further improvement in this research in spite of the work conducted in this research. Therefore, future directions and recommendations of this research are also specified in Chapter 9 towards improvement of reporting and ultimately support for CS in SRs.

CHAPTER 9

Conclusions and Future Work

This chapter presents the summary of the research undertaken in this project and conclusions. Some thoughts on the CS tool are presented alongside suggestions for future work and recommendations for CS practitioners and researchers.

9.1 Summary and conclusions of the research

The overarching goal of this research is to investigate experimental transparency in automatic CS studies. In the course of this investigation, how and what information can aid the reproducibility of the studies and understandability the complexity of the models were explored. A CS tool was also developed as part of the investigation.

The project commenced with a literature review (in the form of a mapping study), the review aimed to identify the methods being used in CS studies and the level of information and justification provided for the methods. The literature review identified a growing field with a moderate number of studies but a lack of technical information that can help in in-depth understanding and interpreting of the performance of the models across the studies. Alongside this finding, the study also identified a lack of reproduction or replication in the research area. The results implied that relying on the studies, independent researchers may be limited to a superficial understanding of the models and thus, find the result hard to reproduce. In addition, the results of the studies have not been independently corroborated and are generally still claims by their authors. These findings provided the motivation for investigating how the information about the studies affected their reproducibility and understanding of their complexity.

33 studies reporting the use of TM for CS were assessed for reproducibility. The study consist three stages: firstly, an attempt was made to actually reproduce the results of six of the studies; secondly, the experience of the reproduction attempt was used to develop a reproducibility assessment framework; and thirdly the framework was used to assess the 33 studies. The assessment focussed on how well the in-

formation provided in the studies can support their reproduction. The results of the work identified information elements of TM for CS experiments essential for their reproduction. The results further identified the information that is often provided and those often overlooked but is critical to reproducing the results of the studies. Based on the experience from the reproducibility assessment, an initial version (version 1.1) of a CS study reproducibility checklist was developed. In an independent but complementary research activity focussing on assessing suitability, brevity and clarity of the checklist items the first refinement of the checklist (version 1.2) was applied to 30 studies. The checklist was further updated to Version 1.3 and validated on nine new studies from the literature review update.

15 healthcare and four SE review datasets were used in the work undertaken to assess structural complexity in CS models. To conduct this study, SVM models whose performance are comparable to those obtainable in existing studies were developed. Two types of feature representations (binary and Word2vec) were explored. The binary feature representation gave acceptable performance only with the non-linear kernel while the Word2vec representation performed well with both the linear and non-linear kernels. In addition to other metrics, the size of the SVs of the SVM models were measured to explore the complexity in the decision making of the models over the datasets. The results of the work identified high levels of complexity in the models, however, the Word2vec representation with the linear kernel models presented complexity lower than their non-linear counterparts while the Word2vec–non-linear kernel models also presented complexity that was relatively low compared to their binary-non–linear kernel counterparts. The results suggest that it is important that model complexity information and how model selection decision are documented in studies.

To explore how model performance can be improved without increasing complexity, the studies used in the complexity assessment were reused with an additional feature - the bibliography data. The feature enrichment study followed that same process as the complexity assessment. But in the feature enrichment study, two sets of models were developed, one with bibliography data added to the features and the other without. The results show that three of the SE datasets that are relatively large showed a consistent performance improvement in terms of recall, precision, MCC and WSS and lower complexity with the the addition of bibliography data. Performance varied with the rest of the datasets however, there were more instances where bibliography enriched datasets exhibited an equal or better performance. Overall, the work finds that the addition of bibliography data to a dataset is more likely to improve the performance and lower complexity of the model than impair it. Though, more investigation is required to further explore the noticed variability.

The methods and findings from the different research activities in this project have been combined and packaged into a tool for automatic CS. The tool was developed

to support the CS stage in SRs and at the same time make enough information about its processes available to support further research for its improvement and progress in the research area.

This research project provides useful insight into the need for more transparency in the reports of CS studies. An in-depth technical understanding of the models is still limited and results are hard to reproduce based on current studies. Though, the results of the studies are promising more independent empirical evidence is still required to verify the results and even improve the quality of the models. Only two of the five tools identified that emerged from some of the studies, have been independently evaluated. More independent evaluation of the tools is also required. The reproducibility assessment checklist and CS tool presented in this thesis aim to support the CS stage of SR by enhancing transparency and facilitating collaborative checking, verification and reproduction of processes and results. The CS tool is particularly developed to serve as a template for future development of transparent tool for CS.

9.2 Future directions for the CS tool

In this section, some potential future refinements for the tool are presented and discussed.

The MVC paradigm used in the development of *TeMACS* ensured that it is easily extendable and adaptable. Easy adaptability of the tool is pertinent given the nascent stage of the research area and the focus of the tool on encouraging independent contribution towards a collaborative solution. It is anticipated that as a result of this research, the field will witness more reproduced work with new solutions and knowledge built on pre-existing ones. As the field's knowledge and solutions evolve, so should the tools to deliver them without necessarily reinventing the wheel. As the field grows, alternative approaches might become suggested, tools might require new features etc., this is why *TeMACS* has been developed ready for adaptation of any of its components with minimal or no effect on the rest.

One key feature the tool will require is to support user collaboration. This feature will give multiple users access to a single project. The tool currently associates projects to a single user. This is sufficient for the purpose of screening citations. But given the fact that a typical SR involves more than one reviewer, it will be good if other reviewers can have access to use the prediction model built for their project.

The tool currently is most useful for repeat reviews where labelled data from the previous review exists or can be recreated. Efforts at improving the relevance and usefulness of this tool will be to incorporate an unsupervised learning approach or semi-supervised learning that will require the labelling of a very small proportion of the dataset and automatically project the class for the rest of the data. Thereby, the

tool will become applicable to new SRs with no pre-existing labelled data.

The current rate of research publications demands that SR are updated constantly than before, hence they become stale quickly. Since documents are static, there should be means to capture SRs and allow them to evolve with time not only in reports but also in process. *TeMACS* currently saves its trained models (for feature representation, feature selection and classification) just to be reapplied on future data. A future refinement will be to take this step further and add the possibility of updated training. That is, after stored models have being used to classify new data, the models' learning will be updated with the features and classes of the new data.

Other researchers are encouraged to use the tool and investigate if the information provided, combined with the description of its operation are sufficient to independently reproduce its results. It would be beneficial if other researchers build upon the tool's idea, recommend refinement or expansion options of the information provided by the tool as well as the reproducibility checklist.

9.3 Recommendations and future work

In this section, recommendations are provided for both SR and CS support researchers.

There had been several studies reporting the use of TM techniques to support the CS stage in SRs. The majority of the studies have used the supervised learning approach combined with the ensemble method. The current trend seems to be shifting towards semi-supervised learning. The SVM remains the most used algorithm due to its versatility with textual data and robustness to imbalanced data classes. The majority of the studies report their experiment process with some assumptions about the reader, which often results in vital information about the studies not being made explicit. There seems to be much concentration on reporting the performance capability of the models and not much on the information that can support the reproduction of the studies and technical understanding to interpret the performance of the models. Reproduced results through study replication have not been witnessed much in the field. Five tools have emerged from the studies, however there has not been much work conducted to validate the tools independently. It may be that some of the tools and methods proposed in other studies are useful for reliable screening of citations in SRs, nevertheless more work need to be undertaken to validate the tools, reproduce the study results and improve the quality of information provided in the studies before any recommendations can be made. To identify the information necessary to be recorded for a study to enhance its reproducibility, researchers are recommended to use the reproducibility *checklist* (more information on the checklist can be found in Section 8.3.1 with background information in Chapter 4) as a guide and use *TeMACS* (see more information in Chapter 7) to automatically collect this

information during the SRs.

CS support researchers are recommended to use and explore *TeMACS* on how reproducibility enabling information can be incorporated into a tool. They are also encouraged to validate the results of the tool independently and contribute the findings as a basis for future developments.

For the purpose of speedy maturation of CS with TM based techniques, reviewers are encouraged (where available) to utilise tools support for their SR tasks and endeavour to give constant feedback on their experiences to the developers. Researchers are also encouraged to utilize the information guidelines provided in the checklist and implemented in *TeMACS* when reporting their studies, evaluating a tool or developing a tool. As the studies continue to increase and more tools emerge, it is suggested that further work to investigate the studies' reproducibility and suitability as well as evaluate the tools be undertaken. The work reported in this thesis has provided a platform for new research in the research area to be undertaken.

Based on the findings of this research a community agreed workshop on possible minimum information guidelines which may be based on the checklist proposed and exemplified in the tool is recommended. This will aid in ensuring the scientific integrity of the literature, improve technical understanding, promote consistency and foster experimental transparency. Further research to investigate the relationship between model complexity and reliability of future prediction, and transparency, researcher's knowledge of TM techniques and reproducibility are suggested. Much of the focus of the investigations and the tool is placed on transparency particularly as it supports reproducibility and understanding the complexity of the models in automatic CS studies using TM techniques. It is suggested that future work be undertaken to investigate other possible factors that may improve evolutionary solution among the studies. Improving the reproducibility of studies may encourage independent researchers to build on existing results which eventually may lead to a timely evolution of a more robust solution. It is anticipated in this research that reproducible studies will strengthen the claim and build more confidence in the proposed method and pave the way for timely advancement. It is therefore suggested that an investigation into the relationship between study reproducibility, the emergence of collaborative solutions and the acceptance rate within the community of the outcome of such collaboration be investigated in the future. The essence of this research is towards the production of viable support for the CS stage in SRs, it would be beneficial to investigate if/how the progress and practices from this research area affect the progress and practices of similar efforts focussed on supporting other SR stages and the whole SR process. It is also suggested to investigate inter-operability among the tools supporting the different SR stages.

References

Aarts, A., Anderson, J., Anderson, C., Attridge, P., Attwood, A., Axt, J., ... Barnett-Cowan, M., et al. (2015). Estimating the reproducibility of psychological science. *Science*, *349*(6251), aac4716.

Abdi, H. & Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, *2*(4), 433–459.

Abu-Mostafa, Y. S. (2012, May). Learning from data. Retrieved from http://work.caltech.edu/slides/slides14.pdf

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

Adeva, J. G. & Calvo, R. (2006). Mining text with pimiento. *IEEE internet computing*, *10*(4), 27–35.

Aggarwal, C. C. (2014). Feature selection for classification: a review. In *Data classification* (pp. 63–90). Chapman and Hall/CRC.

Agosti, M., Di Buccio, E., Ferro, N., Masiero, I., Peruzzo, S., & Silvello, G. (2012). Directions: design and specification of an ir evaluation infrastructure. In *Clef* (pp. 88–99). Springer.

Alvarsson, J., Eklund, M., Andersson, C., Carlsson, L., Spjuth, O., & Wikberg, J. E. (2014). Benchmarking study of parameter variation when using signature fingerprints together with support vector machines. *Journal of chemical information and modeling*, *54*(11), 3211–3217.

Ang, J. C., Mirzal, A., Haron, H., & Hamed, H. N. A. (2016). Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection. *IEEE/ACM transactions on computational biology and bioinformatics*, *13*(5), 971–989.

Babar, M. A. & Zhang, H. (2009). Systematic literature reviews in software engineering: preliminary results from interviews with researchers. In *Empirical software engineering and measurement, 2009. esem 2009. 3rd international symposium on* (pp. 346–355). IEEE.

Baharudin, B., Lee, L. H., & Khan, K. (2010). A Review of Machine Learning Algorithms for Text-Documents Classification. *J. Adv. Inf. Technol. 1*(1), 4–20. doi:10.4304/jait.1.1.4-20

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A., & Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, *16*(5), 412–424.

Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, *39*(3), 930–945.

Barron, A., Rissanen, J., & Yu, B. (1998). The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, *44*(6), 2743–2760.

Bartlett, P. & Shawe-Taylor, J. (1999). Generalization performance of support vector machines and other pattern classifiers. *Advances in Kernel methodssupport vector learning*, 43–54.

Basili, V. R., Shull, F., & Lanubile, F. (1999). Building knowledge through families of experiments. *IEEE Transactions on Software Engineering*, *25*(4), 456–473.

Bekhuis, T. & Demner-Fushman, D. (2010). Towards automating the initial screening phase of a systematic review. In *Medinfo* (pp. 146–150).

Bekhuis, T. & Demner-Fushman, D. (2012). Screening nonrandomized studies for medical systematic reviews: a comparative study of classifiers. *Artificial intelligence in medicine*, *55*(3), 197–207.

Bekhuis, T., Tseytlin, E., Mitchell, K. J., & Demner-Fushman, D. (2014). Feature engineering and a proposed decision-support system for systematic reviewers of medical evidence. *PloS one*, *9*(1), e86277.

Biolchini, J., Mian, P. G., Natali, A. C. C., & Travassos, G. H. (2005). Systematic review in software engineering. *System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES*, *679*(05), 45.

Bird, S. (2006). Nltk: the natural language toolkit. In *Proceedings of the coling/acl on interactive presentation sessions* (pp. 69–72). COLING-ACL '06. Sydney, Australia: Association for Computational Linguistics. doi:10.3115/1225403.1225421

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. K. (1987). Occam's razor. *Information processing letters*, *24*(6), 377–380.

Borah, R., Brown, A. W., Capers, P. L., & Kaiser, K. A. (2017). Analysis of the time and workers needed to conduct systematic reviews of medical interventions using data from the prospero registry. *BMJ open*, *7*(2), e012545.

Bowes, D., Hall, T., & Beecham, S. (2012). Slurp: a tool to help large complex systematic literature reviews deliver valid and rigorous results. In *Proceedings of the 2nd international workshop on evidential assessment of software technologies* (pp. 33–36). ACM.

Breiman, L. et al. (1998). Arcing classifier (with discussion and a rejoinder by the author). *The annals of statistics*, *26*(3), 801–849.

Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of systems and software*, *80*(4), 571–583.

Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, *18*(4), 467–479.

Browne, M. W. (2000). Cross-validation methods. *Journal of mathematical psychology*, *44*(1), 108–132.

Carbonell, J. G., Michalski, R. S., & Mitchell, T. M. (1983). An overview of machine learning. In *Machine learning, volume i* (pp. 3–23). Elsevier.

Carver, J. C., Hassler, E., Hernandes, E., & Kraft, N. A. (2013). Identifying barriers to the systematic literature review process. In *Empirical software engineering and measurement, 2013 acm/ieee international symposium on* (pp. 203–212). IEEE.

Chaitin, G. J. (1969). On the length of programs for computing finite binary sequences: statistical considerations. *Journal of the ACM (JACM)*, *16*(1), 145–159.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Cherkassky, V. & Ma, Y. [Yunqian]. (2004a). Practical selection of svm parameters and noise estimation for svm regression. *Neural Networks*, *17*(1), 113–126. doi:https://doi.org/10.1016/S0893-6080(03)00169-2

Cherkassky, V. & Ma, Y. [Yunqian]. (2004b). Practical selection of svm parameters and noise estimation for svm regression. *Neural networks*, *17*(1), 113–126.

Choi, S., Ryu, B., Yoo, S., & Choi, J. (2012). Combining relevancy and methodological quality into a single ranking for evidence-based medicine. *Information Sciences*, *214*, 76–90.

Chollet, F. et al. (2015). Keras.

Cohen, A. M. (2006). An effective general purpose approach for automated biomedical document classification. In *Amia annual symposium proceedings* (Vol. 2006, p. 161). American Medical Informatics Association.

Cohen, A. M. (2008). Optimizing feature representation for automated systematic review work prioritization. In *Amia annual symposium proceedings* (Vol. 2008, p. 121). American Medical Informatics Association.

Cohen, A. M. (2011). Performance of support-vector-machine-based classification on 15 systematic review topics evaluated with the wss@ 95 measure. *Journal of the American Medical Informatics Association*, *18*(1), 104–104.

Cohen, A. M., Ambert, K., & McDonagh, M. (2009). Cross-topic learning for work prioritization in systematic review creation and update. *Journal of the American Medical Informatics Association*, *16*(5), 690–704.

Cohen, A. M., Ambert, K., & McDonagh, M. (2010). A prospective evaluation of an automated classification system to support evidence-based medicine and systematic review. In *Amia annual symposium proceedings* (Vol. 2010, p. 121). American Medical Informatics Association.

Cohen, A. M., Ambert, K., & McDonagh, M. (2012). Studying the potential impact of automated document classification on scheduling a systematic review update. *BMC medical informatics and decision making*, *12*(1), 33.

Cohen, A. M., Hersh, W. R., Peterson, K., & Yen, P.-Y. (2006). Reducing workload in systematic review preparation using automated citation classification. *Journal of the American Medical Informatics Association*, *13*(2), 206–219.

Cohen, A. M. & Yen, P. Y. (2014). Systematic drug class review gold standard data.

Comeau, D. C., Islamaj Doan, R., Ciccarese, P., Cohen, K. B., Krallinger, M., Leitner, F., ... Torii, M., et al. (2013). Bioc: a minimalist approach to interoperability for biomedical text processing. *Database*, *2013*, bat064.

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

Couban, R. (2016). Covidence and rayyan. *Journal of the Canadian Health Libraries Association/Journal de l'Association des bibliothèques de la santé du Canada*, *37*(3).

Crick, T., Hall, B. A., & Ishtiaq, S. (2014). "Can I Implement Your Algorithm?": A Model for Reproducible Research Software. *ArXiv e-prints*. eprint: 1407.5981

Cristianini, N. & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

Da Silva, F. Q., Santos, A. L., Soares, S., França, A. C. C., Monteiro, C. V., & Maciel, F. F. (2011). Six years of systematic literature reviews in software engineering: an updated tertiary study. *Information and Software Technology*, *53*(9), 899–913.

Dalal, S. R., Shekelle, P. G., Hempel, S., Newberry, S. J., Motala, A., & Shetty, K. D. (2013). A pilot study using machine learning and domain knowledge to facilitate comparative effectiveness review updating. *Medical Decision Making*, *33*(3), 343–355.

Davison, A. (2012). Automated capture of experiment context for easier reproducibility in computational research. *Computing in Science & Engineering*, *14*(4), 48–56.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*(6), 391.

Dietterich, T. G. (2002). Ensemble learning. *The handbook of brain theory and neural networks*, *2*, 110–125.

Domingos, P. (1999a). Metacost: a general method for making classifiers cost-sensitive. In *Proceedings of the fifth acm sigkdd international conference on knowledge discovery and data mining* (pp. 155–164). ACM.

Domingos, P. (1999b). The role of occam's razor in knowledge discovery. *Data mining and knowledge discovery*, *3*(4), 409–425.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78–87.

Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S., & Harshman, R. (1988). Using latent semantic analysis to improve access to textual information. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 281–285). Acm.

Dyba, T., Dingsoyr, T., & Hanssen, G. K. (2007). Applying systematic reviews to diverse study types: an experience report. In *Empirical software engineering and measurement, 2007. esem 2007. first international symposium on* (pp. 225–234). IEEE.

Dyba, T., Kitchenham, B. A., & Jorgensen, M. (2005). Evidence-based software engineering for practitioners. *IEEE software*, *22*(1), 58–65.

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., ... Bouras, A. (2014). A survey of clustering algorithms for big data: taxonomy and empirical analysis. *IEEE transactions on emerging topics in computing*, *2*(3), 267–279.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, *17*(3), 37.

Felizardo, K. R. [Katia R], Andery, G. F., Paulovich, F. V., Minghim, R., & Maldonado, J. C. (2012). A visual analysis approach to validate the selection review of primary studies in systematic reviews. *Information and Software Technology*, *54*(10), 1079–1091.

Felizardo, K. R. [Katia R], Salleh, N., Martins, R. M., Mendes, E., MacDonell, S. G., & Maldonado, J. C. (2011). Using visual text mining to support the study selection activity in systematic literature reviews. In *Empirical software engineering and measurement (esem), 2011 international symposium on* (pp. 77–86). IEEE.

Felizardo, K. R. [Katia Romero], Souza, S. R., & Maldonado, J. C. (2013). The use of visual text mining to support the study selection activity in systematic literature reviews: a replication study. In *Replication in empirical software engineering research (reser), 2013 3rd international workshop on* (pp. 91–100). IEEE.

Fernández-Sáez, A. M., Bocco, M. G., & Romero, F. P. (2010). Slr-tool: a tool for performing systematic literature reviews. In *Icsoft (2)* (pp. 157–166).

Ferro, N. (2017). Reproducibility challenges in information retrieval evaluation. *Journal of Data and Information Quality (JDIQ)*, *8*(2), 8.

Fiszman, M., Bray, B. E., Shin, D., Kilicoglu, H., Bennett, G. C., Bodenreider, O., & Rindflesch, T. C. (2010). Combining relevance assignment with quality of the

evidence to support guideline development. *Studies in health technology and informatics*, *160*(0 1), 709.

Fiszman, M., Ortiz, E., Bray, B. E., & Rindflesch, T. C. (2008). Semantic processing to support clinical guideline development. In *Amia annual symposium proceedings* (Vol. 2008, p. 187). American Medical Informatics Association.

Fodor, I. K. (2002). *A survey of dimension reduction techniques*. Lawrence Livermore National Lab., CA (US).

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, *3*(Mar), 1289–1305.

Fox, C. (1989). A stop list for general text. In *Acm sigir forum* (Vol. 24, *1-2*, pp. 19–21). ACM.

Freire, J., Fuhr, N., & Rauber, A. (2016). Reproducibility of data-oriented experiments in e-science (dagstuhl seminar 16041). In *Dagstuhl reports* (Vol. 6, *1*). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Freund, Y., Schapire, R. E. et al. (1996). Experiments with a new boosting algorithm. In *Icml* (Vol. 96, pp. 148–156). Bari, Italy.

Frunza, O., Inkpen, D., & Matwin, S. (2010). Building systematic reviews using automatic text classification techniques. In *Proceedings of the 23rd international conference on computational linguistics: posters* (pp. 303–311). Association for Computational Linguistics.

Frunza, O., Inkpen, D., Matwin, S., Klement, W., & Oblenis, P. (2011). Exploiting the systematic review protocol for classification of medical abstracts. *Artificial intelligence in medicine*, *51*(1), 17–25.

Gates, A., Johnson, C., & Hartling, L. (2018). Technology-assisted title and abstract screening for systematic reviews: a retrospective evaluation of the abstrackr machine learning tool. *Systematic reviews*, *7*(1), 45.

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., ... Gentry, J., et al. (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome biology*, *5*(10), R80.

Ghafari, M., Saleh, M., & Ebrahimi, T. (2012). A federated search approach to facilitate systematic literature review in software engineering. *International Journal of Software Engineering & Applications*, *3*(2), 13.

Goecks, J., Nekrutenko, A., & Taylor, J. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome biology*, *11*(8), R86.

González-Barahona, J. M. & Robles, G. (2012). On the reproducibility of empirical software engineering studies based on data retrieved from development repositories. *Empirical Software Engineering*, *17*(1–2), 75–89. doi:10.1007/s10664--011--9181--9

Grünwald, P. (2000). Model selection based on minimum description length. *Journal of Mathematical Psychology*, *44*(1), 133–152.

Hady, M. F. A. & Schwenker, F. (2013). Semi-supervised learning. In *Handbook on neural information processing* (pp. 215–239). Springer.

Hafner, R. & Riedmiller, M. (2011). Reinforcement learning in feedback control. *Machine learning*, *84*(1-2), 137–169.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, *11*(1), 10–18.

Hall, T., Beecham, S., Bowes, D., Gray, D., & Counsell, S. (2012). A systematic literature review on fault prediction performance in software engineering. *IEEE Transactions on Software Engineering*, *38*(6), 1276–1304.

Hansen, M. H. & Yu, B. (2001). Model selection and the principle of minimum description length. *Journal of the American Statistical Association*, *96*(454), 746–774.

Hashimoto, K., Kontonatsios, G., Miwa, M., & Ananiadou, S. (2016). Topic detection using paragraph vectors to support active learning in systematic reviews. *Journal of biomedical informatics*, *62*, 59–65.

Hassler, E., Carver, J. C., Kraft, N. A., & Hale, D. (2014). Outcomes of a community workshop to identify and rank barriers to the systematic literature review process. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering* (p. 31). ACM.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, *13*(4), 18–28.

Hernandes, E., Zamboni, A., Fabbri, S., & Thommazo, A. D. (2012). Using gqm and tam to evaluate start-a tool that supports systematic review. *CLEI Electronic Journal*, *15*(1), 3–3.

Hersh, W. (2005). Evaluation of biomedical text-mining systems: lessons learned from information retrieval. *Briefings in bioinformatics*, *6*(4), 344–356.

Higgins, J. P. & Green, S. (2011). *Cochrane handbook for systematic reviews of interventions*. John Wiley & Sons.

Hotho, A., Nürnberger, A., & PaaSS, G. (2005). A brief survey of text mining. In *Ldv forum* (Vol. 20, *1*, pp. 19–62).

Hothorn, T., Held, L., & Friede, T. (2009). Biometrical journal and reproducible research. *Biometrical Journal*, *51*(4), 553–555.

Howard, B. E., Phillips, J., Miller, K., Tandon, A., Mav, D., Shah, M. R., ... Rooney, A. A., et al. (2016). Swift-review: a text-mining workbench for systematic review. *Systematic reviews*, *5*(1), 87. doi:10.1186/s13643-016-0263-z

Hu, J., Fang, L., Cao, Y., Zeng, H.-J., Li, H., Yang, Q., & Chen, Z. (2008). Enhancing text clustering by leveraging wikipedia semantics. In *Proceedings of the*

31st annual international acm sigir conference on research and development in information retrieval (pp. 179–186). ACM.

Ikonomakis, M., Kotsiantis, S., & Tampakas, V. (2005). Text classification using machine learning techniques. *WSEAS transactions on computers*, *4*(8), 966–974.

Inzalkar, S. & Sharma, J. (2015). A survey on text mining-techniques and application. *International Journal of Research In Science & Engineering*, *24*, 1–14.

Ioannidis, J. P., Allison, D. B., Ball, C. A., Coulibaly, I., Cui, X., Culhane, A. C., ... Jurman, G., et al. (2009). Repeatability of published microarray gene expression analyses. *Nature genetics*, *41*(2), 149–155.

Jain, D. & Singh, V. (2018). Feature selection and classification systems for chronic disease prediction: a review. *Egyptian Informatics Journal*. doi:https://doi.org/10.1016/j.eij.2018.03.002

Japkowicz, N. (2000). The class imbalance problem: significance and strategies. In *Proc. of the intl conf. on artificial intelligence*.

Japkowicz, N. (2013). Assessment metrics for imbalanced learning. *Imbalanced learning: Foundations, algorithms, and applications*, 187–206.

Japkowicz, N. & Shah, M. (2011). *Evaluating learning algorithms: a classification perspective*. Cambridge University Press.

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In *European conference on machine learning* (pp. 137–142). Springer.

Jolliffe, I. T. (2002). Principal component analysis and factor analysis. *Principal component analysis*, 150–166.

Jonnalagadda, S. R., Goyal, P., & Huffman, M. D. (2015). Automating data extraction in systematic reviews: a systematic review. *Systematic reviews*, *4*(1), 78.

Jonnalagadda, S. R. & Petitti, D. (2013). A new iterative method to reduce workload in systematic review process. *International journal of computational biology and drug design*, *6*(1-2), 5–17.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: a survey. *Journal of artificial intelligence research*, *4*, 237–285.

Khabsa, M., Elmagarmid, A., Ilyas, I., Hammady, H., & Ouzzani, M. (2016). Learning to identify relevant studies for systematic reviews using random forest and external information. *Machine Learning*, *102*(3), 465–482.

Khan, A., Baharudin, B., Lee, L. H., & Khan, K. (2010). A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, *1*(1), 4–20.

Kim, S. & Choi, J. (2012). Improving the performance of text categorization models used for the selection of high quality articles. *Healthcare informatics research*, *18*(1), 18–28.

Kitchenham, B. A. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, *33*(2004), 1–26.

Kitchenham, B. A., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology*, *51*(1), 7–15.

Kitchenham, B. A. & Brereton, P. (2013). A systematic review of systematic review process research in software engineering. *Information and software technology*, *55*(12), 2049–2075.

Kitchenham, B. A., Budgen, D., & Brereton, P. (2015). *Evidence-based software engineering and systematic reviews*. CRC Press.

Kitchenham, B. A. & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. In *Technical report, ver. 2.3 ebse technical report. ebse*. sn.

Kitchenham, B. A., Dyba, T., & Jorgensen, M. (2004). Evidence-based software engineering. In *Proceedings of the 26th international conference on software engineering* (pp. 273–281). IEEE Computer Society.

Kitchenham, B. A., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering–a tertiary study. *Information and Software Technology*, *52*(8), 792–805.

Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: a survey. *The International Journal of Robotics Research*, *32*(11), 1238–1274.

Kolmogorov, A. N. (1965). Three approaches to the quantitative definition ofinformation'. *Problems of information transmission*, *1*(1), 1–7.

Kontonatsios, G., Brockmeier, A. J., Przybya, P., McNaught, J., Mu, T., Goulermas, J. Y., & Ananiadou, S. (2017). A semi-supervised approach using label propagation to support citation screening. *Journal of biomedical informatics*, *72*, 67–76.

Korde, V. & Mahender, C. N. (2012). Text classification and classifiers: a survey. *International Journal of Artificial Intelligence & Applications*, *3*(2), 85.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: a review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*, 3–24.

Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2006). Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, *26*(3), 159–190.

Kouznetsov, A. & Japkowicz, N. (2010). Using classifier performance visualization to improve collective ranking techniques for biomedical abstracts classification. In *Canadian conference on artificial intelligence* (pp. 299–303). Springer.

Kouznetsov, A., Matwin, S., Inkpen, D., Razavi, A. H., Frunza, O., Sehatkar, M., ... OBlenis, P. (2009). Classifying biomedical abstracts using committees of clas-

sifiers and collective ranking techniques. In *Canadian conference on artificial intelligence* (pp. 224–228). Springer.

Kubat, M., Matwin, S. et al. (1997). Addressing the curse of imbalanced training sets: one-sided selection. In *Icml* (Vol. 97, pp. 179–186). Nashville, USA.

Kumar, A. A. & Chandrasekhar, S. (2012). Text data pre-processing and dimensionality reduction techniques for document clustering. *International Journal of Engineering Research & Technology (IJERT)*, *1*(5), 2278–0181.

Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. (2015). From word embeddings to document distances. In *International conference on machine learning* (pp. 957–966).

Lebanon, G., Mao, Y., & Dillon, J. (2007). The locally weighted bag of words framework for document representation. *Journal of Machine Learning Research*, *8*(Oct), 2405–2441.

Leopold, E. & Kindermann, J. (2002). Text categorization with support vector machines. how to represent texts in input space? *Machine Learning*, *46*(1-3), 423–444.

Liu, J., Timsina, P., & El-Gayar, O. (2016). A comparative analysis of semi-supervised learning: the case of article selection for medical systematic reviews. *Information Systems Frontiers*, 1–13.

Ma, Y. [Yimin]. (2007). *Text classification on imbalanced data: application to systematic reviews automation* (Doctoral dissertation, University of Ottawa (Canada)).

Malheiros, V., Hohn, E., Pinho, R., & Mendonca, M. (2007). A visual text mining approach for systematic reviews. In *Empirical software engineering and measurement, 2007. esem 2007. first international symposium on* (pp. 245–254). IEEE.

Marshall, C. & Brereton, P. (2013). Tools to support systematic literature reviews in software engineering: a mapping study. In *Empirical software engineering and measurement, 2013 acm/ieee international symposium on* (pp. 296–299). IEEE.

Marshall, C. & Brereton, P. (2015). Systematic review toolbox: a catalogue of tools to support systematic reviews. In *Proceedings of the 19th international conference on evaluation and assessment in software engineering* (p. 23). ACM.

Marshall, C., Brereton, P., & Kitchenham, B. A. (2014). Tools to support systematic reviews in software engineering: a feature analysis. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering* (p. 13). ACM.

Marsland, S. (2015). *Machine learning: an algorithmic perspective*. CRC press.

Martinez, D., Karimi, S., Cavedon, L., & Baldwin, T. (2008). Facilitating biomedical systematic reviews using ranked text retrieval and classification. In *Australasian document computing symposium (adcs)* (pp. 53–60).

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, *405*(2), 442–451.

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. *Journal of the American Medical Informatics Association*, *17*(4), 446–453.

Matwin, S., Kouznetsov, A., Inkpen, D., Frunza, O., & O'blenis, P. (2011). Performance of svm and bayesian classifiers on the systematic review classification task. *Journal of the American Medical Informatics Association*, *18*(1), 104–105.

Matwin, S. & Sazonova, V. (2012). Direct comparison between support vector machine and multinomial naive bayes algorithms for medical abstract classification. *Journal of the American Medical Informatics Association*, *19*(5), 917–917.

McKibbon, K. (1998). Evidence-based practice. *Bulletin of the Medical Library Association*, *86*(3), 396.

Menardi, G. & Torelli, N. (2014). Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, *28*(1), 92–122.

Meng, J., Lin, H., & Yu, Y. (2011). A two-stage feature selection method for text categorization. *Computers & Mathematics with Applications*, *62*(7), 2793–2800.

Mesirov, J. P. (2010). Accessible reproducible research. *Science*, *327*(5964), 415–416.

Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (2013). *Machine learning: an artificial intelligence approach*. Springer Science & Business Media.

Miguel, E., Camerer, C., Casey, K., Cohen, J., Esterling, K. M., Gerber, A., ... Imbens, G., et al. (2014). Promoting transparency in social science research. *Science*, *343*(6166), 30–31.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Miller, J. (2005). Replicating software engineering experiments: a poisoned chalice or the holy grail. *Information and Software Technology*, *47*(4), 233–244.

Miner, G., Elder IV, J., & Hill, T. (2012). *Practical text mining and statistical analysis for non-structured text data applications*. Academic Press.

Generalization error of SVM. (2007, September). Online; accessed 17/10/2017. Retrieved from https://ocw.mit.edu/courses/mathematics/18-465-topics-in-statistics-statistical-learning-theory-spring-2007/lecture-notes/l4.pdf

Miwa, M., Thomas, J., OMara-Eves, A., & Ananiadou, S. (2014). Reducing systematic review workload through certainty-based screening. *Journal of biomedical informatics*, *51*, 242–253.

Mo, Y., Kontonatsios, G., & Ananiadou, S. (2015). Supporting systematic reviews using lda-based document representations. *Systematic reviews*, *4*(1), 172. doi:10.1186/s13643-015-0117-0

Molléri, J. S. & Benitti, F. B. V. (2015). Sesra: a web-based automated tool to support the systematic literature review process. In *Proceedings of the 19th international conference on evaluation and assessment in software engineering* (p. 24). ACM.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Myung, I. J. (2000). The importance of complexity in model selection. *Journal of mathematical psychology*, *44*(1), 190–204.

Nannen, V. (2010). A short introduction to model selection, kolmogorov complexity and minimum description length (mdl). *arXiv preprint arXiv:1005.2364*.

Niyogi, P. & Girosi, F. (1999). Generalization bounds for function approximation from scattered noisy data. *Advances in Computational Mathematics*, *10*(1), 51–80.

Olofsson, H., Brolund, A., Hellberg, C., Silverstein, R., Stenström, K., Österberg, M., & Dagerhamn, J. (2017). Can abstract screening workload be reduced using text mining? user experiences of the tool rayyan. *Research synthesis methods*, *8*(3), 275–280.

Olorisade, B. K., Brereton, P., & Andras, P. (2017a). Reporting statistical validity and model complexity in machine learning based computational studies. In *Proceedings of the 21st international conference on evaluation and assessment in software engineering* (pp. 128–133). ACM.

Olorisade, B. K., Brereton, P., & Andras, P. (2017b, August). Reproducibility in machine learning-based studies: an example of text mining. In *Reproducibility in machine learning workshop at the 34th international conference on machine learning*. Sydney, Australia.

Olorisade, B. K., Brereton, P., & Andras, P. (2017c). Reproducibility of studies on text mining for citation screening in systematic reviews: evaluation and checklist. *Journal of biomedical informatics*, *73*, 1–13.

Olorisade, B. K., de Quincey, E., Brereton, P., & Andras, P. (2016). A critical analysis of studies that address the use of text mining for citation screening in systematic reviews. In *Proceedings of the 20th international conference on evaluation and assessment in software engineering* (p. 14). ACM.

Olorisade, B. K., Vegas, S., & Juristo, N. (2013). Determining the effectiveness of three software evaluation techniques through informal aggregation. *Information and Software Technology*, *55*(9), 1590–1601.

O'Mara-Eves, A., Thomas, J., McNaught, J., Miwa, M., & Ananiadou, S. (2015). Using text mining for study identification in systematic reviews: a systematic review of current approaches. *Systematic reviews*, *4*(1), 5.

on Evidence-Based Practice, A. P. T. F. et al. (2006). Evidence-based practice in psychology. *The American Psychologist*, *61*(4), 271.

Opitz, D. W. & Maclin, R. (1999). Popular ensemble methods: an empirical study. *J. Artif. Intell. Res.(JAIR)*, *11*, 169–198.

Ouzzani, M., Hammady, H., Fedorowicz, Z., & Elmagarmid, A. (2016). Rayyana web and mobile app for systematic reviews. *Systematic reviews*, *5*(1), 210.

Paulovich, F. V. & Minghim, R. (2006). Text map explorer: a tool to create and explore document maps. In *Information visualization, 2006. iv 2006. tenth international conference on* (pp. 245–251). IEEE.

Paynter, R., Bañez, L. L., Berliner, E., Erinoff, E., Lege-Matsuura, J., Potter, S., & Uhl, S. (2016). Epc methods: an exploration of the use of text-mining software in systematic reviews.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Dubourg, V., et al. (2011). Scikit-learn: machine learning in python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

Peng, R. D. (2011). Reproducible research in computational science. *Science*, *334*(6060), 1226–1227. doi:10.1126/science.1213847.Reproducible

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic mapping studies in software engineering. In *Ease* (Vol. 8, pp. 68–77).

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, *14*(3), 130–137.

Radjenovi, D., Heriko, M., Torkar, R., & ivkovi, A. (2013). Software fault prediction metrics: a systematic literature review. *Information and Software Technology*, *55*(8), 1397–1418.

Rathbone, J., Hoffmann, T., & Glasziou, P. (2015). Faster title and abstract screening? evaluating abstrackr, a semi-automated online screening program for systematic reviewers. *Systematic reviews*, *4*(1), 80.

Razavi, A. H., Matwin, S., Inkpen, D., & Kouznetsov, A. (2009). Parameterized contrast in second order soft co-occurrences: a novel text representation technique in text mining and knowledge extraction. In *Data mining workshops, 2009. icdmw'09. ieee international conference on* (pp. 471–476). IEEE.

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. In *Encyclopedia of database systems* (pp. 532–538). Springer.

Reynolds, S. (2008). *Evidence-based practice: a critical appraisal*. John Wiley & Sons.

Riaz, M., Sulayman, M., Salleh, N., & Mendes, E. (2010). Experiences conducting systematic reviews from novices' perspective. In *Proceedings of the 19th international conference on evaluation and assessment in software engineering*.

Rissanen, J. (1983). A universal prior for integers and estimation by minimum description length. *The Annals of statistics*, 416–431.

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

Rokach, L. (2005). Ensemble methods for classifiers. In *Data mining and knowledge discovery handbook* (pp. 957–980). Springer.

Rung, J. & Brazma, A. (2013). Reuse of public genome-wide gene expression data. *Nature Reviews Genetics*, *14*(2), 89–99.

Sackett, D. L., Rosenberg, W. M., Gray, J. M., Haynes, R. B., & Richardson, W. S. (1996). Evidence based medicine: what it is and what it isn't. British Medical Journal Publishing Group.

Saha, T. K., Ouzzani, M., Hammady, H. M., & Elmagarmid, A. K. (2016). A large scale study of svm based methods for abstract screening in systematic reviews. *arXiv preprint arXiv:1610.00192*.

Salton, G. & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, *24*(5), 513–523.

Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013). Ten simple rules for reproducible computational research. *PLoS Comput Biol*, *9*(10), e1003285.

Schaffer, C. (1993). Selecting a classification method by cross-validation. *Machine Learning*, *13*(1), 135–143.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, *34*(1), 1–47.

Sharma, K., Sharma, A., Joshi, D., Vyas, N., & Bapna, A. (2017). A review of text mining techniques & applications. *International Journal of Computer (IJC)*, *24*(1), 170–176.

Shemilt, I., Simon, A., Hollands, G. J., Marteau, T. M., Ogilvie, D., O'Mara-Eves, A., ... Thomas, J. (2014). Pinpointing needles in giant haystacks: use of text mining to reduce impractical screening workload in extremely large scoping reviews. *Research Synthesis Methods*, *5*(1), 31–49.

Shull, F. J., Carver, J. C., Vegas, S., & Juristo, N. (2008). The role of replications in empirical software engineering. *Empirical software engineering*, *13*(2), 211–218.

Shull, F., Mendoncça, M. G., Basili, V., Carver, J., Maldonado, J. C., Fabbri, S., ... Ferreira, M. C. (2004). Knowledge-sharing issues in experimental software engineering. *Empirical Software Engineering*, *9*(1), 111–137.

Small, K., Wallace, B., Trikalinos, T., & Brodley, C. E. (2011). The constrained weight space svm: learning with ranked features. In *Proceedings of the 28th international conference on machine learning (icml-11)* (pp. 865–872).

Solomonoff, R. J. (1964). A formal theory of inductive inference. part i. *Information and control*, *7*(1), 1–22.

Srividhya, V. & Anitha, R. (2010). Evaluating preprocessing techniques in text categorization. *International journal of computer science and application*, *47*(11), 49–51.

Staples, M. & Niazi, M. (2007). Experiences using systematic review guidelines. *Journal of Systems and Software*, *80*(9), 1425–1437.

Steinwart, I. (2003). Sparseness of support vector machines. *Journal of Machine Learning Research*, *4*(Nov), 1071–1105.

Sun, Y., Yang, Y., Zhang, H., Zhang, W., & Wang, Q. (2012). Towards evidence-based ontology for supporting systematic literature review.

Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: a review. *Data classification: Algorithms and applications*, 37.

Tax, D. M., Van Breukelen, M., Duin, R. P., & Kittler, J. (2000). Combining multiple classifiers by averaging or by multiplying? *Pattern recognition*, *33*(9), 1475–1485.

Thomas, J. & OMara, A. (2011). How can we find relevant research more quickly. *NCRM MethodsNews. UK: NCRM*, *3*.

Timsina, P., Liu, J., & El-Gayar, O. (2015). Active learning for the automation of medical systematic review creation. In *Twenty-first americas conference on information systems*.

Timsina, P., Liu, J., & El-Gayar, O. (2016). Advanced analytics for the automation of medical systematic reviews. *Information Systems Frontiers*, *18*(2), 237–252.

Timsina, P., Liu, J., El-Gayar, O., & Shang, Y. (2016, January). Using semi-supervised learning for the creation of medical systematic review: an exploratory analysis. In *System sciences (hicss), 2016 49th hawaii international conference on* (pp. 1195–1203). IEEE. doi:10.1109/HICSS.2016.151

Tomassetti, F., Rizzo, G., Vetro, A., Ardito, L., Torchiano, M., & Morisio, M. (2011). Linked data approach for selection process automation in systematic reviews. In *Evaluation & assessment in software engineering (ease 2011), 15th annual conference on* (pp. 31–35). IET.

Tsafnat, G., Glasziou, P., Choong, M. K., Dunn, A., Galgani, F., & Coiera, E. (2014). Systematic review automation technologies. *Systematic reviews*, *3*(1), 74.

Turner, M., Kitchenham, B. A., Budgen, D., & Brereton, P. (2008). Lessons learnt undertaking a large-scale systematic literature review. In *Ease*.

Uuz, H. (2011). A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. *Knowledge-Based Systems*, *24*(7), 1024–1032.

Uysal, A. K. & Gunal, S. (2014). The impact of preprocessing on text classification. *Information Processing & Management*, *50*(1), 104–112.

Vegas, S., Juristo, N., Moreno, A., Solari, M., & Letelier, P. (2006). Analysis of the influence of communication between researchers on experiment replication. In *Proceedings of the 2006 acm/ieee international symposium on empirical software engineering* (pp. 28–37). ACM.

Verma, M., Srivastava, M., Chack, N., Diswar, A. K., & Gupta, N. (2012). A comparative study of various clustering algorithms in data mining. *International Journal of Engineering Research and Applications (IJERA)*, *2*(3), 1379–1384.

Wahono, R. S. (2015). A systematic literature review of software defect prediction: research trends, datasets, methods and frameworks. *Journal of Software Engineering*, *1*(1), 1–16.

Wallace, B. C., Small, K., Brodley, C. E., Lau, J., Schmid, C. H., Bertram, L., ... Trikalinos, T. A. (2012). Toward modernizing the systematic review pipeline in genetics: efficient updating via data mining. *Genetics in medicine*, *14*(7), 663–669.

Wallace, B. C., Small, K., Brodley, C. E., Lau, J., & Trikalinos, T. A. (2010). Modeling annotation time to reduce workload in comparative effectiveness reviews. In *Proceedings of the 1st acm international health informatics symposium* (pp. 28–35). ACM.

Wallace, B. C., Small, K., Brodley, C. E., Lau, J., & Trikalinos, T. A. (2012). Deploying an interactive machine learning system in an evidence-based practice center: abstrackr. In *Proceedings of the 2nd acm sighit international health informatics symposium* (pp. 819–824). ACM.

Wallace, B. C., Small, K., Brodley, C. E., & Trikalinos, T. A. (2010). Active learning for biomedical citation screening. In *Proceedings of the 16th acm sigkdd international conference on knowledge discovery and data mining* (pp. 173–182). ACM.

Wallace, B. C., Small, K., Brodley, C. E., & Trikalinos, T. A. (2011). Who should label what? instance allocation in multiple expert active learning. In *Proceedings of the 2011 siam international conference on data mining* (pp. 176–187). SIAM.

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*(1), 55.

Wang, P. & Domeniconi, C. (2008). Building semantic kernels for text classification using wikipedia. In *Proceedings of the 14th acm sigkdd international conference on knowledge discovery and data mining* (pp. 713–721). ACM.

Wang, S., Mathew, A., Chen, Y., Xi, L., Ma, L., & Lee, J. (2009). Empirical analysis of support vector machine ensemble classifiers. *Expert Systems with applications*, *36*(3), 6466–6476.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., ... Bourne, P. E., et al. (2016). The fair guiding principles for scientific data management and stewardship. *Scientific data*, *3*, 160018.

Wilkinson, M. D., Verborgh, R., da Silva Santos, L. O. B., Clark, T., Swertz, M. A., Kelpin, F. D., ... Ciccarese, P., et al. (2017). Interoperability and fairness through a novel combination of web technologies. *PeerJ Computer Science*, *3*, e110.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann.

Xu, L., Krzyzak, A., & Suen, C. Y. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE transactions on systems, man, and cybernetics*, *22*(3), 418–435.

Yang, P., Hwa Yang, Y., B Zhou, B., & Y Zomaya, A. (2010). A review of ensemble methods in bioinformatics. *Current Bioinformatics*, *5*(4), 296–308.

Yang, Y. & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Icml* (Vol. 97, pp. 412–420).

Yu, W., Clyne, M., Dolan, S. M., Yesupriya, A., Wulf, A., Liu, T., ... Gwinn, M. (2008). Gapscreener: an automatic tool for screening human genetic association literature in pubmed using the support vector machine technique. *BMC bioinformatics*, *9*(1), 205.

Yu, Z., Kraft, N. A., & Menzies, T. (2016). How to read less: better machine assisted reading methods for systematic literature reviews. *CoRR*, *abs/1612.03224*. arXiv: 1612.03224. Retrieved from http://arxiv.org/abs/1612.03224

Zhang, H. & Babar, M. A. (2013). Systematic reviews in software engineering: an empirical investigation. *Information and Software Technology*, *55*(7), 1341–1354.

Zhu, X. (2006). Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, *2*(3), 4.

Zobel, J., Webber, W., Sanderson, M., & Moffat, A. (2011). Principles for robust evaluation infrastructure. In *Proceedings of the 2011 workshop on data infrastructures for supporting information retrieval evaluation* (pp. 3–6). ACM.

APPENDIX A

Excluded Papers

The set of papers excluded from the mapping study discussed in section 3.4.1 is presented in Table A.1 below.

Table A.1: List of excluded papers

S/N	Paper ID	Paper Title	Paper Reference
1	P07	Performance of support-vector-machine-based classification on 15 systematic review topics evaluated with the WSS@95 measure	A. M. Cohen (2011)
2	P16	Combining relevance assignment with quality of the evidence to support guideline development	Fiszman et al. (2010)
3	P17	Semantic processing to support clinical guideline development	Fiszman, Ortiz, Bray, and Rindflesch (2008)
4	P28	Direct comparison between support vector machine and multinomial naive Bayes algorithms for medical abstract classification	Matwin and Sazonova (2012)
5	P31	Parameterized contrast in second order soft co-occurrences: A novel text representation technique in text mining and knowledge extraction	Razavi, Matwin, Inkpen, and Kouznetsov (2009)
6	P33	Towards evidence-based ontology for supporting Systematic Literature Review	Sun, Yang, Zhang, Zhang, and Wang (2012)
7	P36	Performance of support-vector-machine-based classification on 15 systematic review topics evaluated with the WSS@95 measure	Matwin, Kouznetsov, Inkpen, Frunza, and O'blenis (2011)
8	P44	How can we find relevant research more quickly?	Thomas and OMara (2011)
9	P35	An extension of the systematic literature review process with visual text mining: a case study on software engineering	Not found

APPENDIX B

Explanation of Terms in Reproducibility Study

B.1 Tags in Table 4.7

U(Usable for reproduction): This option was used if the information provided for a certain element are precise and was useful to repeat the study action. This is normally associated with a combination of 'complete' tag in 'identification' and 'description'; and 'public' in 'availability' attributes.
D (Usable for reproduction with some difficulty): Any variation in the identification, description and public attributes from the description above will likely result in a 'D' measure if the information was still found useful. For example, if a data source was precisely described but it was stored on a private repository requiring certain membership or the researcher had to take some personal initiative to achieve the expected task.
N (Not usable for reproduction): This indicated the case when the information provided does not help the reader in any way to repeat the author's action(s).
+ (Future availability is foreseeable): This sign is used to indicate that a concrete artefact e.g. tool or dataset will still be available in foreseeable future. May be because it's open source, well maintained, funded, managed or because it's been around for some time with an active team and technical support etc.
* (Flexible): The asterisk sign is used to indicate perceived level of flexibility of:

i) Data: In terms of storage or format. The ease of the possibility to transform it from one format or storage technology to another.

ii) Tools, algorithms or techniques: Was the method or tool written in a popular language with codes made available to the public and easy to modify and/or extend?

- (Irrelevant): Used when an attribute is irrelevant to a given element.
The tags are an overall decision on how useful to reproducibility was the information provided in the study being assessed regarding each information element and

its attribute rating. Table 4.6 provides an example of the attributes judgement per information element for a sample study. In the table, data source has an assessment of 'D+', the 'D' simply implies that the information regarding the data source given in the study being assessed was found useful (i.e. a reader can use it to find the data) but with some level of difficulty (e.g. the link given was to a general page and the reader have to figure out how to navigate to the specific data webpage). The '+' implies that the data was likely to be persistent may be because it's hosted in a public well maintained website or provided by a reputable body that is interested to continue in available.

B.2 Model parameters

The parameter settings for the SVM and the perceptron models presented in Section 4.2.1.4 are shown in the Tables B.1 and B.2. Other parameters not shown for either algorithm are left at their default values.

Table B.1: SVM parameters settings

Parameter	Value
C	1.0
class_weight	'balanced'

B.3 Some terms/phrases in Table 4.8

Following are the definitions of some of the phrases used in Table 4.8:
Raw dataset: This refers to the whole body of the dataset in its original form, in situations where the study under review utilized only a subset of a larger data body. For example, the TREC 2004 dataset consists of 50 DERP review topics where some of the studies reviewed in this study used only 15 or at most 24. The raw dataset in this case is the complete 50 review topics because they were bundled together. Any user will first have to download the whole set before extracting the part required. This

Table B.2: Perceptron parameters settings

Parameter	Value
penalty	'l1'
class_weight	'balanced'
shuffle	True
random_state	0

may sometimes be the same as the target dataset when the whole set is being used.
Target dataset: The target dataset is the subset (data) of interest in its original form, for a particular study in cases where the data used for the study is part of a larger set. An example is the 15 review topics used in A. M. Cohen et al., 2006 which is a subset of the 50 review topics of the TREC 2004 dataset. This may sometimes be the same as the raw dataset.
Cleaned dataset: This is the processed (through preprocessing or any other data cleaning approach) version of the target dataset.
Internal structure: This entry requires the researcher to describe the different headings under which each data record was categorized and which part is of interest to the study. For example, the TREC 2004 used 50 or more categorical heading to describe each document, part of which are: Title, Abstract, MeSH tag, PMID, publication type, publication year etc. The storage format and order of heading arrangement might also be useful.
Data retrieval method: Information about how the dataset was packaged or stored and what method was used or will be required to gain access to the data e.g direct download from a universal resource locator (URL) or automated retrieval (e.g. web scraping) because the dataset are not bundled together or are from different sources.
Data extraction: Most of the data files are sometimes too large to be opened directly or loaded into memory at once, so, after gaining access to the raw dataset, how were the records of interest for each datum extracted. This is more useful in cases where only partial record of each datum is desired. Again, using the TREC 2004 dataset as an example, most of the studies reviewed were interested only in four information - title, abstract, MeSH and the publication type out of about 50 information available for each document.
Custom algorithm: In situations where a researcher proposed a new or an improvement to an existing algorithm, the type of description provided for this proposal will determine how well or not it can be reused.

APPENDIX C

Reproducibility Information

Following are information to support the reproduction of studies reported in Sections 5.4 and 6.3.

i) Initial dataset shuffle seed: 29

ii) StratifiedKfold seed: 37, 71, 21, 61, 55

iii) SVM parameters:

   a) Gamma: auto
   b) C: 1, 10, 100, 1000, 10000
   c) Kernel: rbf, linear, sigmoid
   d) Model random state: 37, 71, 21, 61, 55
   e) Sample weight: 1:4
   f) Class weight: balanced, None

iv) Word2Vec model

   a) Features: as in Table 5.2.
   b) minimum word count: 10
   c) context window: 15

Table C.1: Software information

S/N	Software packages	Version
1	Python	2.7.12 64bit
2	Ipython	5.1.0
3	Scipy	0.18.1
4	Numpy	1.11.3
5	Sklearn	0.18.1
6	Pandas	0.19.2
7	NLTK	3.2.2
8	Gensim	1.0.1
9	Matplotlib	1.5.3

APPENDIX D

TeMACS - Design and Development Details

The design and some of the development details of *TeMACS* is presented in this chapter as a supplement to the discussion of the tool presented in Chapter 7.

D.1 TeMACS features

The user management module of the tool is presented in this section.

D.1.1 Managing a user profile

The user profile management section of the application consists of four functionalities:

i) register - create user account

ii) login - provide users with access to user area of the application

iii) password reset - enable users to recover lost/forgotten password

iv) logout - the logout module performs house cleaning operations, particularly by deleting files saved for temporary use (notably the user's dataset) and log the user out of the application's user area.

D.1.1.1 Register

To begin using the system, the user must be successfully logged into the application. The home page presents a link in text (Figure 7.1) for new users to register with the application. Users are able to register to create an account in the application by providing:

i Name - The user's name

ii email: The email of the user for communication purpose

iii password: A unique password for the protection of the user's profile

iv organization: The user's affiliation

When the *Register* button (Figure D.1) is submitted, it sends a Hypertext Transfer Protocol (http) post request to the *new_user* controller which queries and the *name, email, password, and organization* fields in the *reviewers table* of the database, ensure the user profile is unmatched by any existing record, create the user and sends (through http response) a success message and returns to the *login* view or a failure message to the *register* view. A class diagram of the back-end database is presented in Figure D.2.

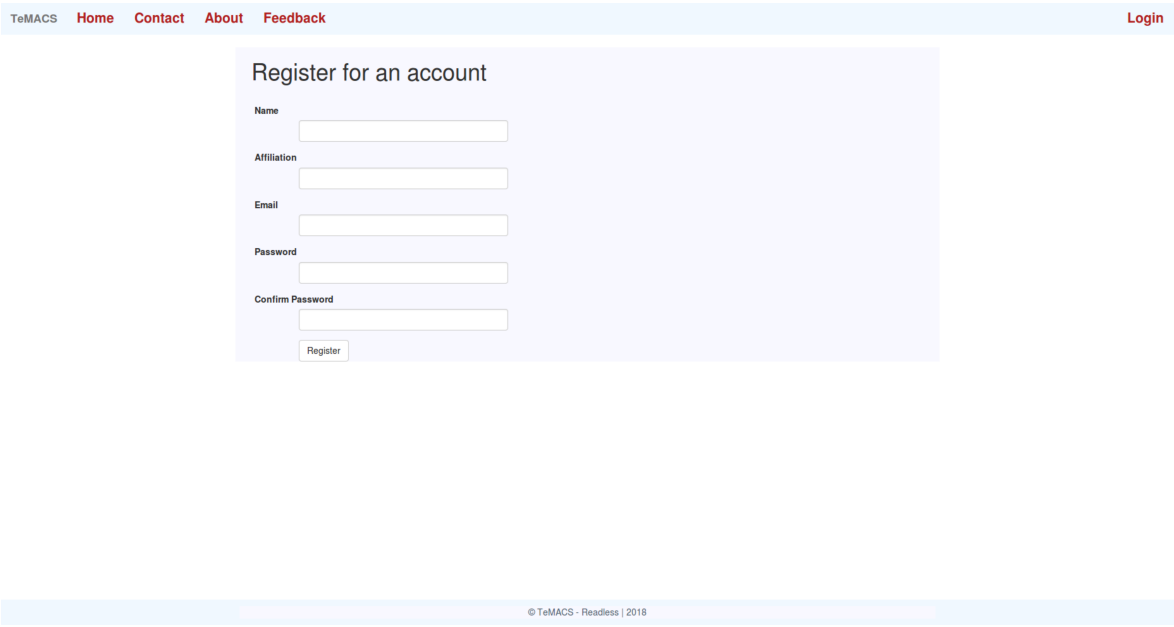

Figure D.1: Screenshot to register new user

D.1.1.2 Login

Registered users are able to log into (from the top right corner, link in Figure 7.1 or a redirection from a successful new user registration) the user area to use the application. A use-case for the user login is presented in Figure D.3. The *login* form contain *username* (email) and *password* fields to login, a password reset option and an option to register if new User (Figure D.4).

Submitting the *login* form sends a http post request to the *login* controller which queries the *username* and *password* fields of the *reviewers* table, verify the particulars match stored values and log the user in or sends error message. The controller sends a success back to and log the user into the user application home - the *dashboard* view (Figure D.5), or sends a failure message back to the *login* view.

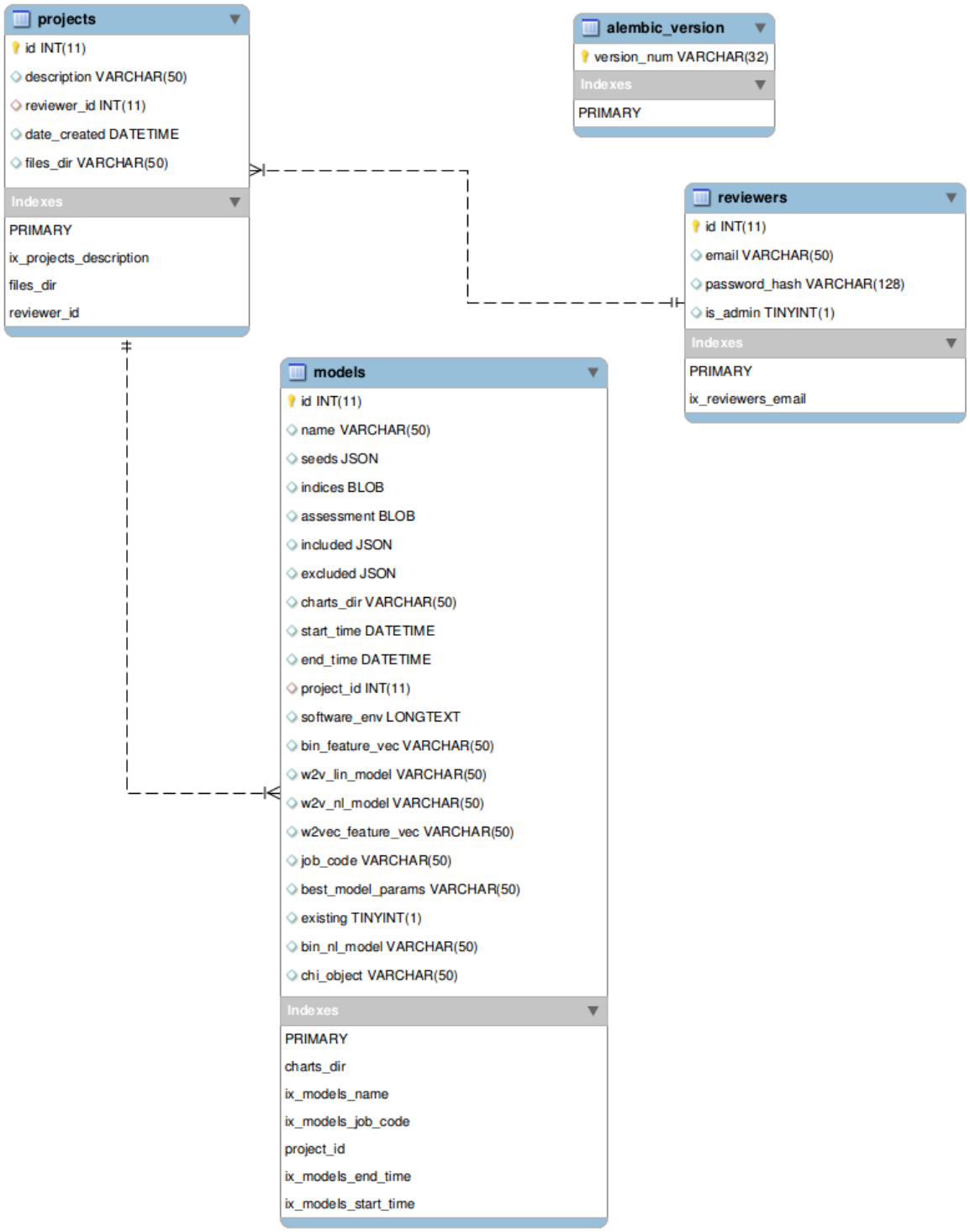

Figure D.2: ER diagram for the application

D.1.1.3 Password reset

Users are able to reset lost or forgotten passwords, in the process a time bound (24 hours) token is generated with a link sent to the user's email to reset their passwords.

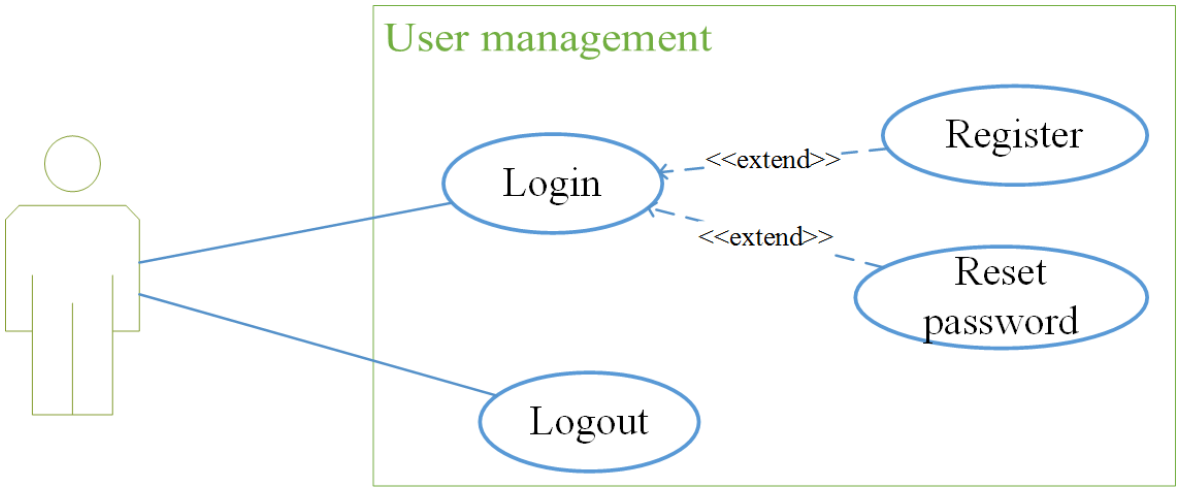

Figure D.3: Login use-case

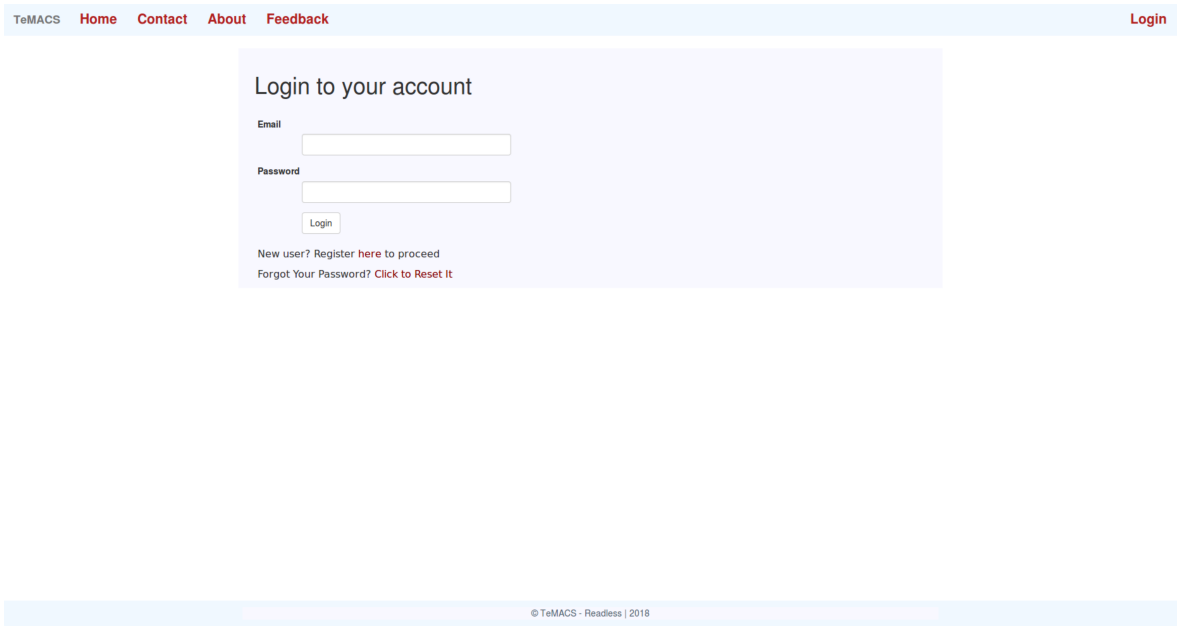

Figure D.4: TeMACS login page

D.1.2 Architecture for the background tasks

The architecture and technologies for running tasks in the background is presented in Figure D.7 below.

Figure D.5: TeMACS dashboard

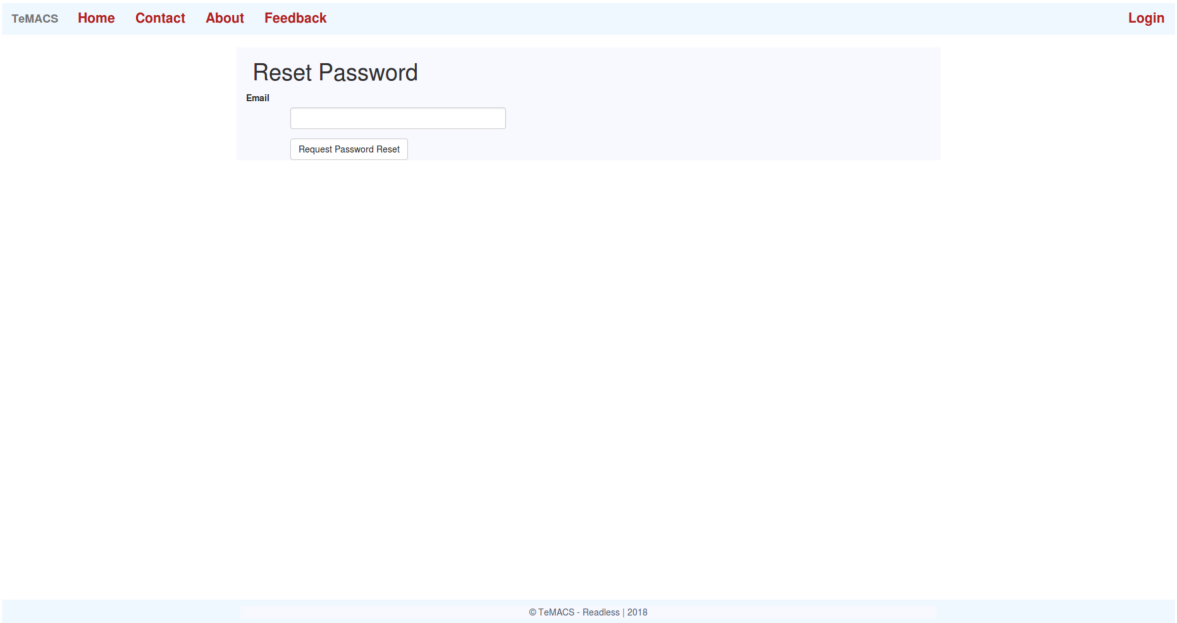

Figure D.6: Screenshot for requesting new password

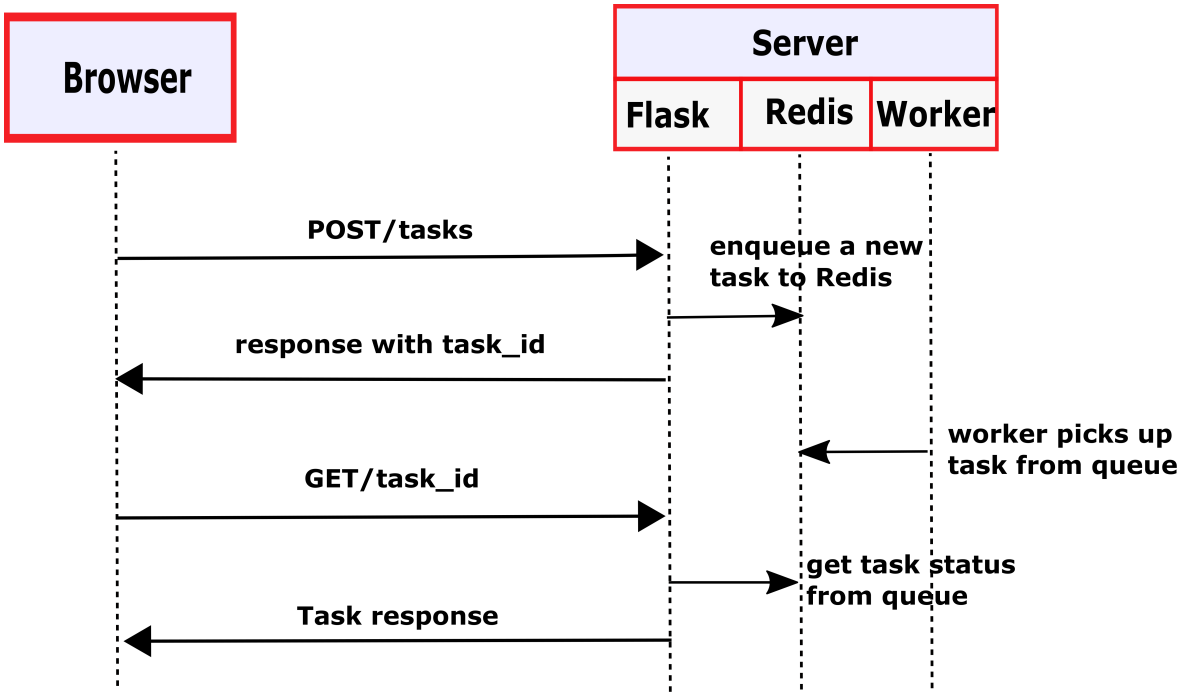

Figure D.7: Architecture for running the classification process in the background