Role of Semantic Indexing for Text Classification

Sadiq Sani

A thesis submitted in partial fulfilment
of the requirements of
Robert Gordon University
for the degree of Doctor of Philosophy

September 2014

Abstract

The Vector Space Model (VSM) of text representation suffers a number of limitations for text classification. Firstly, the VSM is based on the Bag-Of-Words (BOW) assumption where terms from the indexing vocabulary are treated independently of one another. However, the expressiveness of natural language means that lexically different terms often have related or even identical meanings. Thus, failure to take into account the semantic relatedness between terms means that document similarity is not properly captured in the VSM. To address this problem, semantic indexing approaches have been proposed for modelling the semantic relatedness between terms in document representations. Accordingly, in this thesis, we empirically review the impact of semantic indexing on text classification. This empirical review allows us to answer one important question: *how beneficial is semantic indexing to text classification performance*. We also carry out a detailed analysis of the semantic indexing process which allows us to identify reasons why semantic indexing may lead to poor text classification performance. Based on our findings, we propose a semantic indexing framework called Relevance Weighted Semantic Indexing (RWSI) that addresses the limitations identified in our analysis. RWSI uses relevance weights of terms to improve the semantic indexing of documents.

A second problem with the VSM is the lack of supervision in the process of creating document representations. This arises from the fact that the VSM was originally designed for unsupervised document retrieval. An important feature of effective document representations is the ability to discriminate between relevant and non-relevant documents. For text classification, relevance information is explicitly available in the form of document class labels. Thus, more effective document vectors can be derived in a supervised manner by taking advantage of available class knowledge. Accordingly, we investigate approaches for utilising class knowledge for supervised indexing of documents. Firstly, we demonstrate how the RWSI framework can be utilised for assigning supervised weights to terms for supervised document indexing. Secondly, we present an approach called Supervised Sub-Spacing ($S3$) for supervised semantic indexing of documents.

A further limitation of the standard VSM is that an indexing vocabulary that consists only of terms from the document collection is used for document representation. This is based on the assumption that terms alone are sufficient to model the meaning of text documents. However for certain classification tasks, terms are insufficient to adequately model the semantics needed for accurate document classification. A solution is to index documents using semantically rich concepts. Accordingly, we present an event extraction framework called Rule-Based Event Extractor (RUBEE) for identifying and utilising event information for concept-based indexing of incident reports. We also demonstrate how certain attributes of these events e.g. negation, can be taken into consideration to distinguish between documents that describe the occurrence of an event, and those that mention the non-occurrence of that event.

keywords: Semantic Indexing, Text Classification, Semantic Relatedness, Supervised Semantic Indexing, Supervised Indexing, Sentiment Classification, Event Extraction

Acknowledgments

All praises are due to Allah Almighty for making all things possible. I would sincerely like to thank my principal supervisor, Dr. Nirmalie Wiratunga, for all the support, guidance, and motivation I received throughout my PhD. I would also like to thank my secondary supervisors Dr. Stewart Massie and Dr. Robert Lothian for their assistance and guidance. I am grateful to IDEAS and SICSA for the sponsorship and for making my research experience a rich and beneficial one. I would like to thank my friends and colleagues at the IDEAS research institute for their camaraderie and also for assisting me at one time or the other. I would like to thank the IDEAS administration for facilitating a smooth and successful research journey.

I would also like to thank my family for their unwavering love, encouragement and support.

Declarations

I declare that I am the sole author of this thesis and that all verbatim extracts contained in the thesis have been identified as such and all sources of information have been specifically acknowledged in the bibliography.

Parts of the work presented in this thesis have appeared in the following publications:

Chapter 3

- Sani, S., Wiratunga, N., Massie, S., Lothian, R.: Should Term-Relatedness be Used in Text Representation. In: Proc. of the 22nd International Conference on Case-Based Reasoning, ICCBR (2013)
- Sani, S., Wiratunga, N., Massie, S., Lothian, R.: When to Generalise - A Case-Based Approach to Text Modelling. In: Proc. of the 17th UK Workshop on Case- Based Reasoning, UKCBR (2012)

Chapter 4

- Sani, S., Wiratunga, N., Massie, S., Lothian, R.: Term similarity and weighting framework for text representation. In: Proc. of the 19th International Conference on Case-Based Reasoning, ICCBR (2011)

Chapter 6

- Sani, S., Wiratunga, N., Massie, S., Lothian, R.: Sentiment Classification Using Supervised Sub-Spacing. To appear in: Proc. of SGAI International Conference on Artificial Intelligence, BCS SGAI (2013)

Chapter 7

- Sani, S., Wiratunga, N., Massie, S., Lothian, R.: Event extraction for reasoning with text. In: Proc. of the 20th International Conference on Case-Based Reasoning, ICCBR (2012)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The explosion of user generated content on the Web has produced significant interest in deriving valuable information and insights from text in order to support intelligent decision making by business organisations as well as governments. Text mining is the research discipline that is concerned with the discovery and utilising of information from unstructured text. The applications of text mining today include: information management, customer experience management, marketing, business intelligence, security, and healthcare, to name just a few. An important area of text mining is text classification. Text classification is the task of categorising unstructured text documents into one or more predefined categories. The significance of text classification is set to increase due to the fact that many text mining tasks can be framed directly as text classification tasks, or rely on text classification as an important intermediate step. We include a list of some of the more popular applications of text classification below.

- Document Management and Retrieval: Modern document management and retrieval systems commonly index documents belonging to different topics separately. This is in order to allow learning of user interests and to support personalisation. Text classification is employed to classify new documents automatically into the defined categories.

- Message Filtering and Organisation: Categorisation of emails is important for maintaining an organised inbox and comes standard in many modern email client systems. The most basic email classification is to categorise emails as Spam and Non Spam. However, some email clients also categorise in-coming mail into Social and Promotion for social media and promotional emails respectively. Examples of emails for each category are relatively easy to

collect which provides opportunity for using text classification to automatically categorise emails.

A Similar idea is also applicable for organising the timeline of micro-blogging sites e.g. Twitter. The velocity of messages (called tweets) flowing into a user's timeline can be difficult to keep track of. However, organising these tweets by topic can allow the user to easily navigate to important or relevant tweets.

- Opinion Mining: Opinion text is frequently generated in the form of reviews and user comments which provides ample opportunity for marketing and predictive analytics. Opinion mining is typically interested in categorising opinion text into positive and negative opinion categories which can naturally be modelled as a text classification task.

- News Categorisation: News reports are naturally presented to readers in different sections by topic. Thus, news categorisation is a suitable application for text classification, in order to ease the burden of manual categorisation of huge volumes of news reports produced daily.

The datasets used in this thesis (described in Section 2.6) have been chosen to reflect the scenarios discussed above e.g. Ohsumed for document management, 20 Newsgroups for message filtering, Reuters Volume 1 for news categorisation and Movie Reviews for opinion mining.

1.1 Vector Space Model

One of the most widely used models of text representation is the Vector Space Model (VSM) which was originally proposed by Salton (Salton, Wong & Yang 1975) for the task of information retrieval (IR). Since then, the VSM along with its fundamental ideas and concepts has been successfully adopted for text classification. The main idea behind the VSM is to represent documents as vectors in an n-dimensional space of features. The basic set of features that has been traditionally used for text representation in the VSM is the set of all unique terms V, in the document collection, called the vocabulary. The use of terms as features in the VSM is based on 4 main assumptions as follows:

- Given any document, the set of terms in the document capture the meaning or semantics of that document

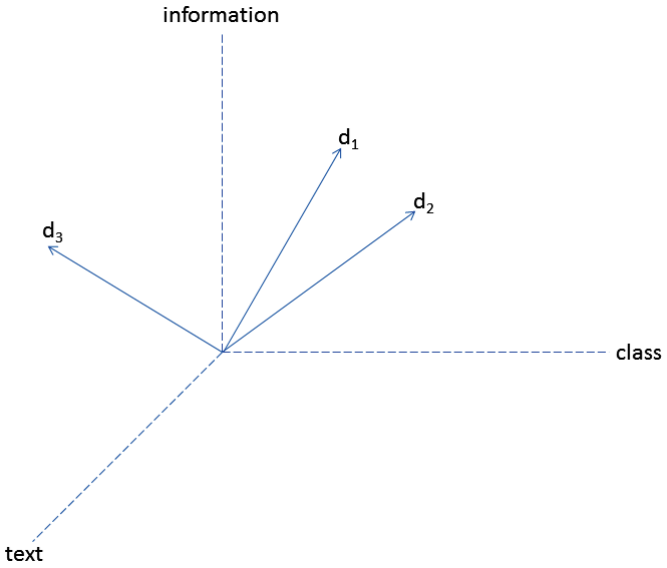

Figure 1.1: Example vector space with three documents and three terms

- The relevance of a term to a document (local relevance) is a function of the frequency of occurrence of the term in that document
- Given any document collection, some terms are more useful for discriminating between relevant and irrelevant documents in that collection (global relevance)
- The global relevance of any term can be measured as a function of the frequency of occurrence of the term in the document collection.

Thus, each term from the vocabulary occupies a separate dimension in the vector space and any given document d_i can be represented as a vector in the space of terms as shown in Equation 1.1.

$$\vec{d_i} = (t_{i,1}, t_{i,2}, ..., t_{i,n}) \tag{1.1}$$

Where $t_{i,j}$ is a combination of the local and global weight of term t_j in document d_i and represents how much t_j contributes to the understanding of the semantics of d_i. $t_{i,j}$ also represents the magnitude of document d_j in the dimension of term t_i in the vector space. Hence, the set of all term weights $t_{i,j}$ of any document d_j provide the exact location d_j in the term-document space.

Figure 1.1 shows a trivial example of a vector space with three terms, *information*, *text* and *class*, and three documents, d_1, d_2 and d_3. The positions of each document in the space is determined by the weights of the vector components of that document along the dimensions of the three

terms. Accordingly, the similarity between any two documents in the space can be computed as a function of the distance between the document vectors in the space. For example from Figure 1.1, d_1 and d_2 are more similar because they are closer to each other than they both are to d_3. From this, it is evident that accurately estimating the similarity between documents in the VSM depends very much on effective weighting of terms in document vectors . Salton (Salton & Buckley 1988) identified three main factors that an effective term-weighting strategy for the VSM should satisfy:

- Relevant documents should be retrieved
- Non-relevant documents should be avoided
- Document length should be normalised

Based on these three factors, the normalised *tf-idf* weighting, which employs a combination of within-document term frequency (local weight) and inverse document frequency (global weight), was introduced (Salton & Buckley 1988). The term frequency component of *tf-idf* estimates the relevance of a term t_i to a document d_j as a function of the frequency of t_i in d_j. This is based on the intuition that the frequency with which a term is used in a document is directly related to the relevance of the term to that document. The inverse document frequency component is designed to assign higher weight to terms that are concentrated in a few documents. This is based on the notion that more specific terms are better at distinguishing the small set of relevant documents from the larger set of irrelevant documents.

There are a few fundamental limitation with Salton's VSM. The first is the assumption of term independence where, the VSM assigns different terms to different dimensions in the vector space with no relationship between these different dimensions. Thus, two documents that do not share identical terms in common will be positioned very distant from each other in the term-document space. However, terms in a vocabulary are not completely independent and often, different terms have very similar or identical meaning. This means that document similarity is not properly captured by the VSM and addressing this problem will require a model of similarity or relatedness between vocabulary terms to be introduced.

A second limitation of Salton's VSM is the complete lack of supervision. Recall that an important consideration when generating document vectors is the ability to distinguish between relevant and non-relevant documents. Because the VSM was originally proposed for unsupervised

document retrieval, the notion of relevance was estimated using inverse document frequency of terms. However, for text classification, relevance information is explicitly provided in the form of document class labels. Therefore, there is opportunity in the case of text classification to take advantage of supervision in order to generate more effective document vectors than would otherwise be produced if class knowledge is ignored.

Thirdly, the VSM represents documents in the space of unique terms from the collection. This is based on the assumption that terms are sufficient to model the meaning of documents. However, this is more of a simplifying argument than a completely accurate one. Indeed, it is well understood that terms often fail to model the right level of semantics needed for accurate document retrieval or classification. Addressing this limitation often requires documents to be represented using more semantically rich concepts as features.

In the next section, we discuss text classification algorithms that are popularly used for document categorisation with the VSM.

1.2 Text Classification Algorithms

Text classification using the VSM involves training a classifier Φ on a collection of training documents D where each document $d_j \in D$ is associated with a class label. Thus, given a new document d_q with unknown class, d_q is represented as a vector $\vec{d_q} = (t_{q,1}, t_{q,2}, ..., t_{q,n})$ in a term-document space. The classifier Φ can now be applied to the vector $\vec{d_q}$ to determine the class membership of document d_q. In the following sub-sections, we describe the two main classification algorithms used with the vector space document representations, *k*-Nearest Neighbour and Support Vector Machines. These two algorithms are also known to produce the best performance on text classification, compared to other classifiers e.g Naive Bayes, Rocchio and C4.5 (Joachims 1998).

1.2.1 *k*-Nearest Neighbour

The *k*-Nearest Neighbour (*k*NN) algorithm is based on the intuition that the class of any given document is likely to be the same as that of the documents most similar to it. Recall that in the VSM, similarity between documents is estimated by the distance between their vector representations is Euclidean space. Thus given any document d_q, the *k*NN classifier would assign to d_q the class of the majority of documents in the neighbourhood of d_q in the term-document space. This

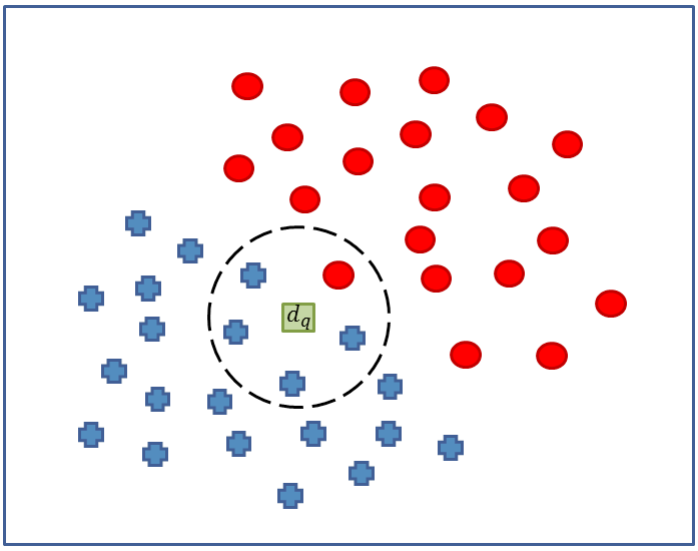

Figure 1.2: Illustration of nearest-neighbour classification of a document d_q.

Metric	Formula
Euclidean	$\sqrt{\sum_i^n (d_{1,i} - d_{2,i})^2}$
Dice	$\frac{2\|d_1 \cap d_2\|}{\|d_1\| + \|d_2\|}$
Jaccard	$\frac{\|d_1 \cap d_2\|}{\|d_1\| + \|d_2\|}$
Cosine	$\frac{\sum_i^n d_{1,i} d_{2,i}}{\|d_1\| \|d_2\|}$

Table 1.1: Document similarity/distance metrics.

is illustrated in Figure 1.2.

An important consideration for *k*NN classification is the similarity metric to use for obtaining the neighbours of d_q. A number of distance and similarity metrics can be used e.g. Euclidean, Dice, Jaccard and Cosine. Given any two document vectors $\vec{d_q}$ and $\vec{d_j}$, a list of these metrics is given in Table 1.1. Note that Dice and Jaccard metrics are not vector distance measures. Rather both metrics measure the similarity between two sets. However we include them here because of their popular use for computing document similarity.

Cosine metric has emerged as the most popular measure of similarity for text documents. Also, comparative evaluations with some of the metrics e.g Euclidean (Chakraborti, Mukras, Lothian, Wiratunga, Watt & Harper 2007), have shown cosine to perform better. The superiority of Cosine over the other metrics can be attributed to a few reasons. Firstly, Euclidean metric measures the absolute distance between two points in space. This means that two vectors that have the same direction (contain the same terms) can have a huge difference computed by Euclidean metric

because the common terms have huge differences in weights (or magnitude).

Secondly both Dice and Jaccard metrics measure similarity between documents as a function of the amount of terms that is shared between them, compared to the amount of terms available in the union of the two documents. This means that documents that have a higher proportion of shared terms are more similar. However, note that the relative weight or importance of the terms is not taken into account. This makes both metrics limited in their application to the VSM where much semantic information is typically encoded in the weights of terms. Accordingly, in the remainder of this thesis, we use cosine similarity metric for identifying the neighbourhood of any given document.

Typically with *k*NN, more than one neighbouring document is considered for deciding the class of the query document d_q. Therefore, it is often the case that the neighbouring documents do not all belong to same class. As shown in Figure 1.2, neighbours of d_q include four documents from the same class (blue cross) and one document from a different class (red dot). In this case, a strategy is required to decide which of the classes to assign to d_q. Given a set E of documents in the neighbourhood of d_q, where each document $d_j \in E$ has corresponding class label c_j and similarity s_j with d_q, two strategies are commonly employed for deciding the class of d_q as follows:

- Majority Voting Strategy: Here, a tally of all documents $d_j \in E$, organised by class, is made and d_q is assigned to the class with the maximum number of documents.

- Similarity Weighted Voting Strategy: A disadvantage of the majority voting strategy is that it assumes all neighbours to be equally important in determining the class of d_q. However, it is intuitive to assume that more similar neighbours should contribute more to the classification decision. Thus, in the similarity weighted strategy, each neighbour contributes to the decision with a weight equivalent to its similarity to d_q. This way, more similar neighbours contribute more to determining the class of d_q than less similar ones. For this reason, we use the similarity weighted voting strategy throughout this thesis.

1.2.2 Support Vector Machines

Support Vector Machines (SVMs) are an example of kernel learning algorithms that have been successfully adopted for text classification (Joachims 1998) using the vector space model. SVMs work for text classification by mapping documents from an original term-document space into a

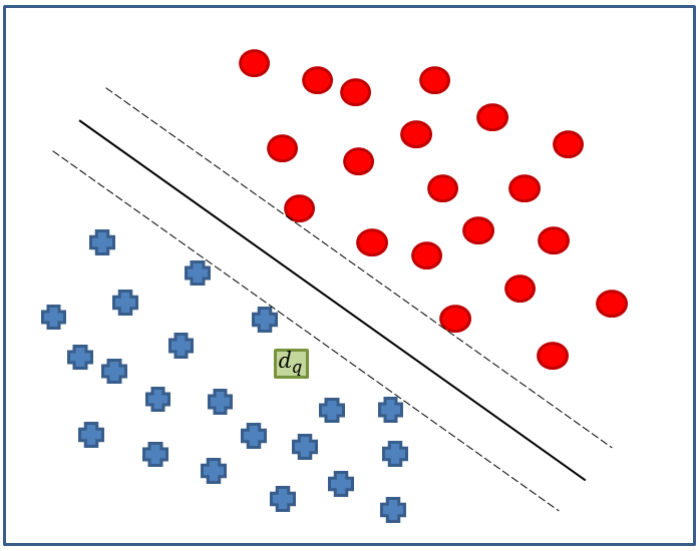

Figure 1.3: Illustration of SVM classification of a document d_q.

higher dimensional space, and then finding the optimal hyperplane in that space that separates, by the widest margin, positive and negative examples of a target class. Thus, SVM is naturally a binary class classifier and requires adaptation for muti-class classification tasks e.g. by running multiple one-vs-all classifications for each class (Duan & Keerthi 2005). Given any new document d_q with unknown class label, an SVM classifier would map d_q to a position on one side of the hyperplane and assign d_q to the class of documents on that side of the hyperplane as shown in Figure 1.3.

SVMs have proven to be very effective for text classification and are often considered to be the state-of-the-art in classifiers. However, other studies suggest instance-based learners such as k Nearest Neighbour (kNN) are equally competitive with SVM performance (Yang & Liu 1999) especially with proper document representation (Colas & Brazdil 2006). In addition, kNN is much simpler to implement than SVM and requires less parameter tuning. kNN also provides good scalability to higher numbers of classes as it naturally supports multi-class classification.

1.3 Research Motivation and Objectives

Text representation using the VSM employs a fundamentally flawed assumption, that terms in a document are independent of one another and thus occupy independent dimensions of the vector space. This assumption was originally included in the design of the VSM in order to simplify

the computation required to work with the model. Accordingly, the more two documents contain different terms, the further away they are from one another in the vector space. However, natural language text is inherently characterised by variety and diversity in word usage which means that different (but semantically related) terms are often used to express the same idea. In the VSM however, documents that contain non-identical but semantically related terms would be positioned far apart in space. This means that the exact lexical match used in the VSM for computing document similarity is not sufficient for estimating the full semantic similarity between documents.

To address the limitation imposed by the term independence assumption, semantic indexing approaches were introduced. Examples of popular semantic indexing approaches include Latent Semantic Indexing (LSI) (Deerwester, Dumais, Landauer, Furnas & Harshman 1990) and the Generalised Vector Space Model (GVSM) (Wong, Ziarko, Raghavan & Wong 1987). These approaches address the problem of term independence by applying semantic transformation operations to document representations in order to reflect the semantic relatedness between vocabulary terms. The effect of semantic indexing on document vectors is that the semantic relatedness between terms becomes encoded in the resulting term weights. While some improvements have been realised using semantic indexing, the benefit has not been consistent. Thus, the primary goal of this thesis is to investigate the performance of semantic indexing on text classification with a view to identifying limitations in the current state-of-the-art. Our goal is then to develop algorithms and techniques that address these limitations.

A second limitation of the VSM for text classification is the lack of supervision in the process of creating document representations. Effective document vectors should be good at distinguishing between relevant and non-relevant documents. Because the VSM was originally designed for unsupervised document retrieval, heuristics such as *idf* were developed in order to approximate the notion of relevance. However, in text classification, relevance information is explicitly provided in the form of class labels. It is therefore intuitive that more effective document vectors can be constructed using supervised approaches that take into account class membership of documents. Accordingly, a second aim of this thesis is to optimise the process of document indexing for text classification by proposing supervised approaches that take advantage of class knowledge.

A further limitation of traditional VSM is that the indexing vocabulary is composed of arbitrary terms from the content of documents. However, for certain types of tasks, a terms-based index is not sufficient for capturing the semantics needed for accurate document classification. Some

approaches have been proposed for addressing this limitation by indexing documents using vectors of concepts defined in a lexicon or an ontology. Note that this approach is also referred to as semantic indexing (Fernandez, Cantador, Lopez, Vallet, Castells & Motta 2011). Unlike the first type of semantic indexing which learns the conceptual structure of text by learning the semantic relatedness of terms, this second type of semantic indexing uses explicit concepts as an indexing vocabulary. Accordingly, a secondary goal of this thesis is to explore the use of semantic indexing that involves the encapsulation semantics at higher levels of abstraction in the VSM. In particular, we will explore the utility of semantically-richer (such as events and entities) indexing vocabulary, over semantically-poor term-based features on text classification tasks.

In order to address the three limitations of the VSM discussed above, this thesis has the following five objectives:

1. Conduct an analysis of the performance of semantic indexing for text classification.
2. Propose a new semantic indexing framework that addresses the limitations identified in 1.
3. Develop a supervised extension of the framework developed in 2 that utilises class knowledge for optimised semantic indexing.
4. Investigate the application of the semantic indexing frameworks developed in 2 and 3 to other classification tasks e.g. sentiment classification.
5. Explore the use of higher level semantic concepts e.g. events for document indexing.

1.4 Contributions

The most significant contribution of this thesis is the development of the Relevance Weighted Semantic Indexing (RWSI) framework which introduces relevance weighting into semantic indexing. Our development of the RWSI framework is based on our discovery that term relevance is essential for effective semantic indexing and that this information is not captured in traditional semantic indexing approaches. A key advantage of the RWSI framework is that it is flexible enough to be used with any semantic relatedness metric and also, any effective term weighting approach.

A second significant contribution is the development of the supervised sub-spacing ($S3$) framework for introducing supervision into semantic indexing. The key idea of $S3$ is to create separate

sub-spaces for each class within which semantic indexing transformations are applied exclusively to documents that belong to that class. In this way, $S3$ is able to modify document representations such that documents that belong to the same class are made more similar to one another.

The third contribution of this thesis is the application of the $S3$ framework to the task of sentiment classification. $S3$ is able to produce document representations that are more effective for sentiment classification by learning semantic relatedness and term weights exclusively from the set of documents belonging to the same sentiment class. Doing so allows $S3$ to emphasise the semantic associations of terms belonging to the same sentiment category in document representations. We further demonstrate how sentiment scores from a sentiment lexicon can be used to further improve the performance of $S3$ on sentiment classification.

Our fourth contribution is a demonstration of the utility of events for document indexing. Accordingly, we present an unsupervised heuristic approach for the extraction of events called RUle-Based Event Extractor (RUBEE). RUBEE uses natural language processing together with a set of rules for extracting events and their attributes from the content of a given text document.

Our final contribution is a detailed evaluation of semantic indexing with semantic relatedness knowledge extracted using both knowledge-resource-based, and distributional approaches. Considering that extracting semantic relatedness is a computationally expensive process, we propose an approach for determining when and when not to apply semantic relatedness using meta-learning.

1.5 Thesis Outline

The rest of this thesis is outlined as follows: In Chapter 2, a review of relevant background and related works is presented. We discuss, in detail, text representation using the VSM and further explain the main limitations of text representation using the VSM for text classification. We discuss a number of semantic indexing and supervised document indexing approaches that have been proposed for addressing these limitations and we analyse the strengths and limitations of these approaches. We conclude Chapter 2 with a discussion of the datasets we use in our experiments and a chapter summary.

In Chapter 3, we empirically review the performance of semantic indexing for text classification by analysing the performance of a variety of semantic indexing approaches on a number

of text classification datasets. Our goal in this chapter is to evaluate whether consistent improvements in text classification performance is realised from semantic indexing. We use our findings from this evaluation to develop a case-based system to recommend, given any dataset, whether or not to employ semantic indexing. We conclude the chapter with an evaluation of our developed case-based system and a discussion of our results.

In Chapter 4, we present a detailed analysis of the semantic indexing process. We demonstrate how relevance information of terms is not captured during semantic indexing which leads to poor text classification performance. Accordingly, we present the Relevance Weighted Semantic Indexing (RWSI) framework which utilises relevance weights of terms for improved semantic indexing of documents. We also demonstrate how the RWSI framework can be utilised exclusively for assigning supervised weights to terms for supervised document indexing.

Semantic indexing is traditionally an unsupervised process. Accordingly, the document representations produced are not optimal for text classification. In Chapter 5 we present a supervised framework called Supervised Sub-Spacing ($S3$) for supervised semantic indexing of documents. $S3$ works by partitioning the term document space into class-based subspaces and applies the RWSI semantic indexing framework to each sub-space independently.

In Chapter 6, we investigate the applicability of our developed semantic indexing approaches to the task of sentiment classification. Sentiment lexicons are commonly used in sentiment classification to provide sentiment scores of terms. Thus, we demonstrate how sentiment scores from a sentiment lexicon can be utilised with the $S3$ framework for improved sentiment classification performance.

In Chapter 7 we present our exploration of semantic indexing using higher level semantic concepts. Accordingly, we present an algorithm called RUBEE for the extraction of event information from the content of incident reports for the purpose of document indexing. We also present a framework for using events for semantic indexing of documents. We further demonstrate how attributes like the polarity (negation) of events can be utilised in the indexing approach. In our evaluation, we compare the RUBEE framework with term-only document indexing using the approach presented in Chapters 4 and 5. Results show our events-based index to lead to better text classification performance compared to term-based indexing.

We conclude this thesis in Chapter 8 with a summary of our main contributions and proposals for future extensions to our work.

Chapter 2

Literature Review

Text representations enable automatic processing of natural language text documents by providing computational models that sufficiently capture the semantics of these documents. However, sufficiently and effectively modelling the semantics of natural language is non-trivial. The VSM has been proposed for the purpose of text representation. However, there are three main problems with the traditional VSM that limit the performance of this model for text classification. These problems are outlined as follows:

- Variation in indexing vocabulary;
- Lack of supervision in document representation; and
- Use of terms only for document indexing

Several approaches have been proposed for addressing the limited semantics of the standard BOW model. One such approach is the use of phrases, rather than individual terms, for document indexing. This is important because multi-term expressions occur often in documents, and these multi-term expressions typically have a meaning that is different to that of the individual terms independently e.g. “machine learning”. Thus, indexing documents using phrases attempts to index documents using such phrases in order to preserve their meaning. The most popular approach identify these phrases statistically by looking for sequences of terms that occur frequently (Caropreso, Matwin & Sebastiani 2000). However, experimental results have not been able to convincingly prove that phrases are useful for text classification (Sebastiani 2002).

Another attempt at overcoming the limitation of the standard BOW for text classification is the use of distributional features (Xue & Zhou 2006). Here, the authors argue that the *compactness* (the spread of the distribution of a term in document) and *position of first appearance* of a term in a document are more important than frequency of appearance. Accordingly, this approach uses a vector V_j to represent a document d_j, where the weight of each term t_i in v_j can be derived by measuring the compactness (CP), position of first appearance (FA) or term frequency (TF) of t_i in d_j. The proposed representation was evaluated using both SVM and kNN classifiers on three datasets. Results show distributional features improved classification performance compared to a standard BOW representation and also, that the performance of distributional features is closely related to the length and writing style of documents. Note however, that the distributional features proposed in (Xue & Zhou 2006) work are not the same as distributional semantic relatedness which is the interest of this work. The key difference is that distributional features propose to replace the frequency counts of terms in the standard VSM with vector representations that capture the position of first appearance of terms and also a measure of the compactness of the appearance of terms in a document. Thus, unlike distributional semantic relatedness approaches, distributional features do not model the semantic relatedness between terms.

While the above two approaches are interesting in their own rights, they do not address the first two problems we outlined. For example, the use of distributional features does not take into account the semantic relatedness between terms and neither does it utilise supervision. The use of phrases for indexing on the other hand only attempts to take into account sequential relationship between terms rather than general semantic relatedness. Thus, a phrasal approach will not take into account the fact that 'coffee' and 'tea' are both types of hot drinks. Accordingly, in this thesis, we are interested in approaches that address the three problems outlined above.

The problem of variation in vocabulary is due to the expressiveness of natural language text which has always presented a big challenge for automated text processing. Natural language text is inherently characterised by variety and diversity in word usage. For example, different terms can be, and often are, used to denote the same thing e.g. the terms 'buy' and 'purchase' are pretty much synonymous and can be used interchangeably. In addition, some words tend to be conceptually similar even though not synonymous. For example the terms 'bus' and 'car' are similar because they are both types of 'motor vehicle'. However, the two words are not synonymous. Words can also share some other types of relationships based on common association e.g. the words

'coal' and 'fire'. In general, these types of relationships between words are referred to as semantic relatedness (Budanitsky & Hirst 2006).

Failure to take into account semantic relatedness between terms leads to problems for automated text processing. This problem is commonly referred to as the vocabulary mismatch in IR literature (Girill 1985). Vocabulary mismatch happens when a set of one or more relevant documents D_r is not retrieved in response to a query q, simply because the documents $d_i \in D_r$ do not contain the exact same terms as the query q. Note that documents in D_r are considered relevant because they contain terms that are semantically related and hence, close in meaning to the terms in q. However, retrieval in the vector space model is based on an exact match between query and document terms and hence, the reason for the failure to retrieve documents in D_r. This same problem transfers to text classification using the VSM. Text classification algorithms such as kNN and SVM also depend on a direct match between terms in the query document d_q and the learned classification model, φ, in order to decide which category to assign to d_q. Thus, addressing this problem requires semantic indexing approaches that attempt to model the semantic relatedness between terms in document representations.

The second problem of lack of supervision in the process of creating document representations arises from the fact that traditional VSM was designed for unsupervised document retrieval. An important goal of document representation in the VSM is to discriminate between relevant and non-relevant documents. Accordingly, the *tf-idf* weighting scheme was introduced (Salton & Buckley 1988) to approximate the notion of relevance in unsupervised document collections. The *tf* component of the weight captures the local relevance of a term to a document as a function of the frequency of the term in that document. On the other hand, the *idf* component captures the global relevance of terms by assigning higher weight to terms that are concentrated in fewer documents. This is based on the notion that more specific terms are better at distinguishing the small set of relevant documents from the larger set of irrelevant documents. However, for text classification, relevance information is explicitly available in the form of class label of documents. Thus, a more effective term weighting scheme can be derived in a supervised manner by taking into account class knowledge. Accordingly, many approaches have proposed applying supervision to document indexing by introducing supervised term weights that are learned using class knowledge.

The third problem arises from the fact that the standard VSM assumes that terms alone are sufficient to model the meaning of text documents. While significant improvements over term-based indexing vocabulary have proved very difficult to achieve (Gomez, Cortizo, Puertas & Ruiz 2004, Castells, Fernandez & Vallet 2007, Mudinas, Zhang & Levene 2012), for certain types of classification tasks, terms are insufficient to adequately model the distinction between relevant and non-relevant documents. The limitations of the keyword indexing vocabulary is usually addressed by indexing documents using concepts either from a lexicon (Gomez et al. 2004) or from an ontology (Kiryakov, Popov, Terziev, Manov & Ognyanoff 2004). This approach to document representation is also referred to as semantic indexing (Fernandez et al. 2011). However, the concepts provided by general purpose lexicons and ontologies do not necessarily have enough coverage to adequately model the semantics of all target domains. On the other hand, building and maintaining domain specific lexicons and ontologies requires significant knowledge engineering effort which makes this a very expensive option. Accordingly, as a secondary objective of this thesis, we explore the use of information extraction for indexing of incident reports using event information extracted directly from the textual content of these reports. This allows us to be able to tackle semantic classification tasks such as filtering out incident documents that report injuries from those that don't.

In the following sections, we present a critical review of works done in the area of semantic relatedness extraction, semantic indexing, supervised document indexing, as well as concept-based indexing, in order to identify the extent to which they address the identified problems with traditional VSM. Our goal is to analyse the strengths and to highlight the limitations of current state-of-the-art approaches, and to propose techniques for addressing these limitations.

2.1 Semantic Relatedness

Addressing the problem of variation in indexing vocabulary requires the use of semantic relatedness metrics, which quantify the degree to which any two given terms are related in meaning (Gracia & Mena 2008). Thus, semantic relatedness is defined in a broad sense to include any type of semantic relationship (Resnik 1995). For example, two terms can have (almost) identical meaning e.g. 'shout' and 'yell', or conceptually similar e.g. the words 'sneakers' and 'boots' both denote types of 'footwear'. Words can also share a semantic relationship based on common asso-

ciation e.g. the words ‘coal’ and ‘fire’, or be related because one is a part of the other (meronymy) e.g ’wheel’ and ’cart’.

More formally, we define semantic relatedness as a function that accepts a pair t_1 and t_2 from a set of terms V and returns a numeric value of how related the two terms are as show in Equation 2.1. Accordingly, the higher the semantic relatedness between any two terms, the stronger the relationship between the terms. We also define semantic relatedness to be bound between the range $\{0, 1\}$. This is done for two reasons. Firstly, this allows for defining an upper bound on semantic relatedness such that identical terms have a semantic relatedness of value of 1 and completely un-related terms have semantic relatedness of 0. Secondly, semantic relatedness is only useful as a relative value on a scale rather than in absolute terms. For example, the fact that the semantic relatedness between ’car’ and ’bus’ is 0.8 is more valuable if we know that this is out of a maximum possible value of 1.0.

$$Rel(t_1, t_2) : V \times V \rightarrow \Re \tag{2.1}$$

The computation of semantic relatedness between terms has many applications that go beyond semantic indexing. Indeed, semantic relatedness computation has its roots in artificial intelligence and psychology, with the works on spreading-activation theory (Quillan 1966, Collins & Loftus 1975). Since then, more work on computing semantic relatedness has been done in the area of natural language processing for applications such as malapropism detection and correction (Budanitsky & Hirst 2006), word sense disambiguation (Patwardhan, Banerjee & Pedersen 2003), lexical selection for automatic machine translation (Wu & Palmer 1994b), multiple choice synonym detection (Turney 2002, Weale, Brew & Fosler-Lussier 2009), and plagiarism detection (Chen, Yeh & Ke 2010).

Semantic relatedness has traditionally been computed using several different approaches. These approaches can be broadly categorised into distributional and knowledge-resource-based as illustrated in figure 2.1. Distributional approaches involve using co-occurrence between terms in a target corpus as a measure of their relatedness. In this way, terms that co-occur more often are judged to be more similar than terms that co-occur less often. Several algebraic functions (e.g. cosine similarity and LSI) and information-theoretic measures (e.g PMI) can be employed for this purpose. On the other hand, knowledge-resource-based approaches employ the aid of a (typically

manually constructed) knowledge resource that contains a sufficient number of terms and relationships between these terms. The structure of these knowledge resources can typically be viewed as a graph where terms are nodes and the relation between terms are edges. This allows for computing the semantic relatedness between terms as a function of the path connecting the two terms in the knowledge resource.

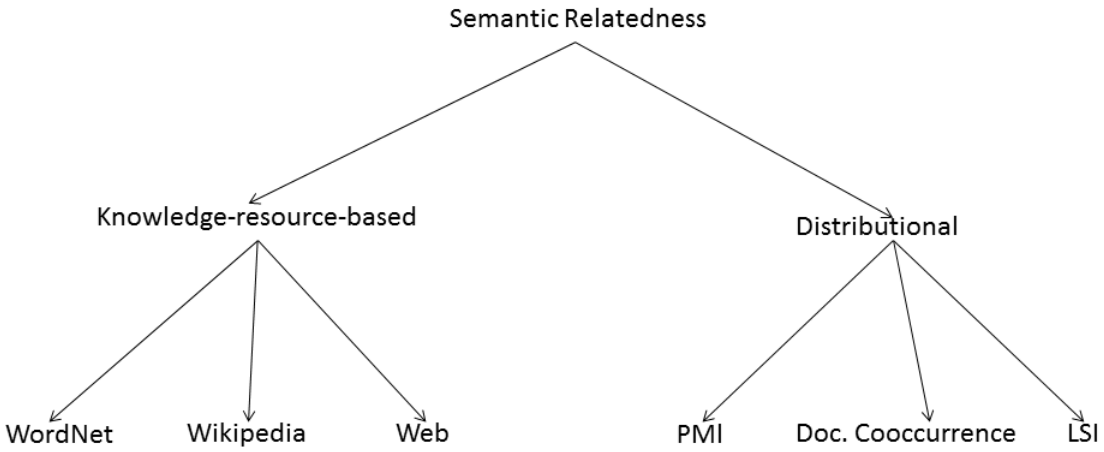

Figure 2.1: Semantic Relatedness.

The most popular knowledge resource used for computing semantic relatedness is the WordNet lexicon (Miller 1995). Terms within WordNet are inter-connected through links representing the semantic and lexical relationships between them. This structure can be viewed as a graph or taxonomy which allows for measuring relatedness between terms by means of combining shortest path between term pairs and information about the depth of nodes in the taxonomy. Another knowledge resource which has recently become very popular is Wikipedia. Similar to WordNet, Wikipedia's category structure can also be viewed as a taxonomy and similar measures used with WordNet can then be adapted for measuring term relatedness using Wikipedia.

Other approaches go beyond Wikipedia and exploit the entire Web as a means for extracting semantic relatedness knowledge e.g. using page counts (Cilibrasi & Vitanyi 2007). Page count of documents returned in response to a search engine query provides useful evidence of relatedness between the terms in the query. This can then be quantified as a semantic relatedness metric i.e. the higher the proportion of documents that contain both terms, the more related the two terms are. However page count can often be misleading as it does not consider the intended context of terms and the semantics within which they are used in the result pages. Sophisticated approaches

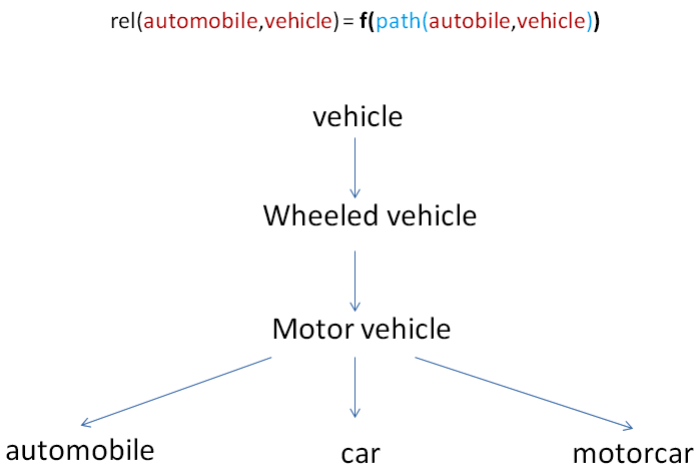

Figure 2.2: Term Relatedness from a taxonomy structure.

using text snippets [1] can be used to improve on page count by exploiting lexico-syntactic patterns in these snippets (Bollegala, Matsuo & Ishizuka 2007).

2.1.1 Knowledge-Resource-Based Approaches

In the following subsections, we describe WordNet and Wikipedia and present techniques for extracting semantic relatedness from these resources.

WordNet

WordNet, is a lexical database for the English language (Miller 1995), which has been used extensively for extracting term-relatedness knowledge. Terms within WordNet are grouped into sets of cognitive synonyms often referred to as concepts. Concepts are further grouped based on their grammatical function into noun, verb, adjective and adverb dictionaries. Concepts within the same dictionary are inter-connected through links representing the semantic and lexical relationships between them. This structure can be viewed as a graph where concepts are nodes and semantic links are edges that form a path between concepts. Hence, computing the semantic relatedness between any two terms t_1 and t_2 using WordNet involves mapping t_1 and t_2 to corresponding concepts c_1 and c_2 in WordNet and then estimating semantic relatedness as a function of the path between c_1 and c_2.

[1] small pieces of text extracted by the search engine around the query term

Many different functions have been applied for computing semantic relatedness using WordNet which include distance-based metrics that compute relatedess as a measure of the shortest path between concept pairs and the depth of nodes in the graph (Wu & Palmer 1994a), and also information theoretic metrics that compute relatedness based on information content (Resnik 1995, Jiang & Conrath 1997, Lin 1998).

The first category of WordNet semantic relatedness measures are the distance-based metrics. The first of these is the Wu and Palmer metric (Wu & Palmer 1994b). Given any two terms t_1 and t_2 which can be mapped to corresponding concepts c_1 and c_2 in WordNet, the semantic relatedness between t_1 and t_2 can be computed using the Wu and Palmer (WUP) metric as follows:

$$Rel_{WUP}(t_1, t_2) = \frac{2 \times depth(lcs(c_1, c_2))}{len(c_1, lcs(c_1, c_2)) + len(c_2, lcs(c_1, c_2)) + 2 \times depth(lcs(c_1, c_2))} \quad (2.2)$$

Where $depth(c)$ is a function that returns the depth of the node c from the global root of the WordNet noun hierarchy and $lsc(c_1, c_2)$ is a function that returns the lowest common subsumer (i.e. lowest common parent concept) of both concepts c_1 and c_2 and $len(c_1, c_2)$ is a function that returns the length of the path connecting c_1 and c_2 in WordNet. In this way, the Wu and Palmer metric measures relatedness between two concepts by calculating the distance between the two concepts to their common ancestor and scaling that distance with the depth of the common ancestor in the taxonomy.

A second distance-based measure is the Leacock and Chodorow metric (Leacock & Chodorow 1998). The semantic relatedness between two terms t_1 and t_2 can be computed using the Leacock and Chodorow (LCH) metric as:

$$Rel_{LCH}(t_1, t_2) = -log \frac{len(c_1, c_2)}{2 \times max\_depth(c)} \quad (2.3)$$

Where $max\_depth(c)$ is the maximum depth of the taxonomy.

A general problem with calculating term relatedness based on the length of the path-length that the length of the path between concepts in a taxonomy may not necessarily reflect the semantic distance between the concepts in real-life. In other words, parts of the WordNet taxonomy are very shallow where concepts are subsumed by very abstract concepts while other parts of the taxonomy are very dense with many subsumption layers. For this reason, it is necessary to utilise approaches

that are able to deal with the problem of disparity in distance between concepts in a taxonomy and their actual semantic distance.

Information content based measures are designed to downplay the importance of the length of links in the taxonomy by utilising additional evidence in the form of corpus statistics. The premise behind these approaches is that relatedness between two concepts c_1 and c_2 can be estimated by the extent to which they share information in common. This can be achieved by determining the information content of the most common concept that subsumes both c_1 and c_2. For any concept c, let $p(c)$ be the probability of the occurrence of c. Then from information theory, the information content of c can be calculated as:

$$IC(c) = -logP(c) \tag{2.4}$$

The probabilities of concepts can be estimated from frequency counts gathered from large corpora such as the one-million-word Brown Corpus of American English. An alternative approach for calculating information content intrinsically from the taxonomy structure without the need for an external corpus was introduced in (Seco, Veale & Hayes 2004). This approach called intrinsic information content (IIC) calculates the information content of a concept based on it's hyponym count such that the more hyponyms a concepts has, the less information it conveys. The formula for calculating IIC is given as:

$$IIC(c) = 1 - \frac{log(hyp(c) + 1)}{logtotal(c)} \tag{2.5}$$

Where $hyp(c)$ returns the number of hyponyms of the concept c and $total(c)$ returns the total number of concepts in the taxonomy. Thus the root concept has an IIC of 0 and a leaf concept has the maximum IIC value of 1.

A number of semantic relatedness metrics have been devised that use information content. The first metric proposed by Lin measures semantic relatedness between two concepts as the ratio of the amount of information needed to describe the commonality between the two concepts to the amount of information needed to describe each concept independently (Lin 1998). The amount of information needed to describe the commonality between two concepts is defined as the information content of the least common subsumer of both concepts. Semantic relatedness

between two terms t_1 and t_2 using the Lin (LIN) metric is defined as follows:

$$Rel_{LIN}(t_1, t_2) = \frac{2 \times IC(lcs(c_1, c_2))}{IC(c_1)IC(c_2)} \tag{2.6}$$

Another semantic relatedness metric that uses information content was proposed by Jiang and Conrath, which measures the difference between the amount of information needed to describe the commonality between two concepts and the amount of information needed to describe each concept independently (Jiang & Conrath 1997). Accordingly, semantic relatedness can be computed using the Jiang and Conrath (JCN) metric as follows:

$$Rel_{JCN}(t_1, t_2) = 2 \times IC(lcs(c_1, c_2)) - IC(c_1)IC(c_2) \tag{2.7}$$

The Jiang and Conrath Metric is by default a distance measure i.e. the higher the the Jiang and Conrath measure between two concepts, the more unrelated the two concepts are. The Jiang and Conrath metrics is converted to similarity measure by taking the inverse.

Despite its popularity, WordNet has recently been criticised for having limited coverage and scope of applications (Gracia & Mena 2008). The implication for semantic indexing is that some terms from the indexing vocabulary may not have corresponding entries in WordNet. This restricts semantic relatedness computation to only the terms that are covered by the WordNet vocabulary which excludes many domain specific terms, abbreviations and slang. WordNet is also known to suffer from sparsity in connections between concepts (Boyd-graber, Fellbaum, Osherson & Schapire 2006). Concepts are typically connected using hierarchical parent-child connections. This means that it is not straightforward to compute the relatedness between related concepts that are not connected through a common parent concept. Also, the different dictionaries within WordNet are independent with very limited inter-connections between them. This means that most metrics are only able to compute semantic relatedness between terms from the same part-of-speech category. This is quite restrictive and does not allow for capturing the full relatedness between terms

Wikipedia

Unlike WordNet, Wikipedia, a free online encyclopedia, boasts vast coverage in orders of magnitude greater than that of lexical databases and thesauri. Wikipedia is particularly attractive as a

source of semantic knowledge because each Wikipedia page provides a comprehensive description of a single topic or concept and can thus be seen as a representation of that concept. Several techniques have been introduced for calculating term relatedness using Wikipedia. A very popular technique is the Explicit Semantic Analysis (ESA) approach presented in (Gabrilovich & Markovitch 2009). This approach attempts to explicitly represent the meaning of natural language by representing text documents in a high-dimensional space of Wikipedia concepts. Let D be a collection of documents where each document d_i is represented using a *tf-idf* vector $\vec{d_i}$ where each entry $v_j \in \vec{d_i}$ is the *tf-idf* weight of word $w_j \in d_i$. Let V be the vocabulary covered by the document collection D. Let C be the collection of all Wikipedia concepts. An inverted index K called a semantic interpreter is created where each vector $\vec{k_j} \in K$ represents the association between the corresponding term w_j and the concepts in C. Thus the semantic interpretation of a document d_i from the term-document space to the concept space is given by

$$\vec{c_i} = \sum_{w_j \in d_i} v_j . \vec{k_j} \tag{2.8}$$

Where $\vec{c_i}$ is the concept vector representation of the document d_i. Thus the similarity between two documents d_i and d_l can be computed by calculating the cosine similarity of their corresponding concept vectors $\vec{c_i}$ and $\vec{c_l}$.

A recent review of the ESA approach revealed that this approach works by exploiting term co-occurrence in Wikipedia (Gottron, Anderka & Stein 2011). The authors also found that using Wikipedia as an index collection for building the semantic interpreter did not perform best i.e. other collections such as the Reuters corpus provided even better results. Furthermore, a semantic index vector with random weights was found to perform nearly as good as the index vector created using Wikipedia pages. This leads to the conclusion that while the ESA approach is effective in improving text retrieval, it is neither taking advantage of, nor exploiting the semantic structure of Wikipedia. In other words, the ESA approach cannot be regarded as a true knowledge-resource-based approach for estimating term relatedness as the semantic interpreter could equally be created from any suitable document collection or corpus with equal or better results.

A second category of Wikipedia-based metrics treat the Wikipedia category structure as a taxonomy. Consequently, existing taxonomy-based metrics like the ones used for WordNet can be adapted to work with the Wikipedia category structure. Such an approach was first intro-

duced in (Strube & Ponzetto 2006) where the authors investigate three categories of term relatedness measures - distance based (Leacock & Chodorow (Leacock & Chodorow 1998) and Wu & Palmer (Wu & Palmer 1994a)), information content based (Resnik (Resnik 1995)), and text overlap based (extended gloss overlap (Banerjee & Pedersen 2003)) metrics.

The results of some studies indicate that Knowledge-resource-based approaches are known to produce estimates of semantic relatedness that more closely match human judgment (Budanitsky & Hirst 2006). However for the purpose of text classification, a more useful estimate of semantic relatedness is one that better reflects the relatedness between terms in the target corpus (Chakraborti, Wiratunga, Lothian & Watt 2007). This is definitely an advantage for distributional approaches which can easily be ported to any specific domain or target corpus. Indeed, this is very much the reason behind the success of techniques such as LSI and LDA that model semantic relatedness that are specific to the underlying document collection. Also, knowledge resources are typically far from being complete and without anomalies. In particular, WordNet is well known for being sparse and having very limited coverage of domain-specific terms (Boydgraber et al. 2006). Also, knowledge resources typically contain multiple senses of the same term. This is particularly worse for Wikipedia where, for example, the word car has over 50 senses including the names of movies, music, sports, people and places. For some terms, the number of senses could easily scale up to several hundreds. This means that the performance of semantic indexing using WordNet and Wikipedia depends on effective mapping of terms to the correct sense within these resources. These reasons make distributional approaches particularly attractive as an effective and efficient option for computing semantic relatedness.

2.1.2 Distributional Approaches

Co-occurrence of terms within a given context in text corpora has been used extensively to infer semantic relatedness. The principal motivation behind using corpus co-occurrence for term relatedness is the distributional hypothesis which states that words that occur in the same context tend to have similar meaning. Thus two terms are similar to the degree to which they co-occur within similar contexts as shown in figure 2.3. Several word contexts have been exploited for obtaining term relatedness. For example, the hyperspace analogue for language (HAL) model introduced in (Lund & Burgess 1996) uses a window of words on either side of the target word as context. A word context derived from syntactic relations is presented in (Padó & Lapata 2007).

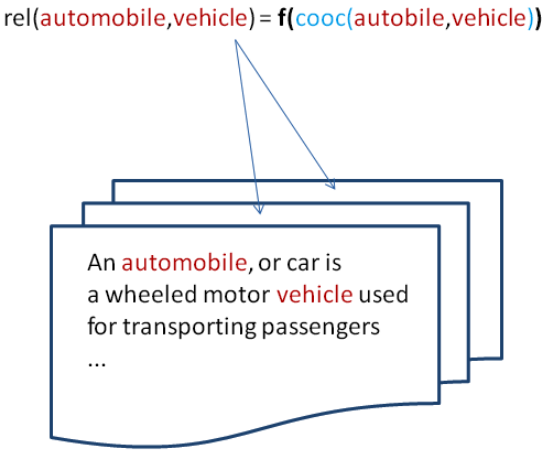

Figure 2.3: Term Relatedness from Corpus Distribution.

Using syntactic relationships rather than simple co-occurrence allows to abstract over word order and to also restrict consideration to associations of a defined semantic type rather than arbitrary co-occurrences. Thus, the algorithm presented in (Padó & Lapata 2007) is designed to construct semantic spaces from text annotated with grammatical relations. A more semantically rich event context for words is presented in (Yan, Maxwell, Song, Hou & Zhang 2010) where sentences are annotated with event information using the PropBank predicate-argument structure (Palmer, Gildea & Kingsbury 2005).

All contexts presented so far have an associated processing cost. For example determining context from window length requires a certain number of words on either side of each target word to be maintained. Also, a context derived from syntactic and semantic relationships requires the text to be annotated with such relationships. A much simpler approach is to consider the entire document as the word context (Deerwester et al. 1990). Thus two words are similar to the extent to which they occur is similar documents. A document context is also preferred in cases where documents in the training corpus have short length. In the following sections we present three different techniques for extracting term relatedness from corpus co-occurrence.

Document Co-occurrence

Documents are considered to be similar in the vector space model (VSM) if they contain a similar set of terms. In the same way, terms can also be considered similar if they appear in a similar set of documents. Given a standard term-document matrix D where columns vectors represent doc-

uments and the row vectors represent terms, the similarity between two terms can be determined by finding the distance between their vector representations. The relatedness between two terms, t_1 and t_2 using the cosine similarity metric is given in equation 2.9.

$$Rel_{DocCooc}(t_1, t_2) = \frac{\sum_{i=0}^{n} t_{1,i} t_{2,i}}{\| t_1 \| \| t_2 \|} \tag{2.9}$$

Latent Semantic Indexing

Recall that LSI uses SVD to exploit co-occurrence patterns of terms and documents to create a semantic concept space which reflects the major associative patterns in the corpus. In this way, LSI brings out the underlying latent semantic structure in texts. Accordingly, the semantic relatedness of these terms can be obtained from the LSI decomposition. Given a term-document matrix D, the decomposition of D is shown in equation 2.10.

$$D = U \times S \times V \tag{2.10}$$

Where U is a term by dimension matrix, S a diagonal matrix of singular values and V a document by dimension matrix. The U, S, V matrices are truncated to k dimensions which represent the k most important concepts in the term-document space. Multiplying the truncated U and S matrices produces rank-reduced term by dimension matrix U' as shown equation 2.11.

$$U' = U \times S \tag{2.11}$$

Semantic relatedness can thus be computed using an approach similar to 2.9 by calculating the cosine similarity of term vectors in U' as shown in equation 2.12.

$$Rel_{LSI}(t_1, t_2) = \frac{\sum_{i=0}^{n} t_{1,i} t_{2,i}}{\| t_1 \| \| t_2 \|} \tag{2.12}$$

Where $t_1 \in U'$ and $t_2 \in U'$.

Normalised Positive Pointwise Mutual Information

The use of mutual information to model term associations is demonstrated in (Church & Hanks 1990). Given two terms t_1 and t_2, mutual information compares the probability of observing

t_1 and t_2 together in a given context (in this thesis we use a document level context), with the probability of observing them independently as shown in equation 2.13.

$$PMI(t_1, t_2) = log_2 \frac{P(t_1, t_2)}{P(t_1)P(t_2)} \tag{2.13}$$

If a significant association exists between between t_1 and t_2, then the joint probability $P(t_1, t_2)$ will be much larger than the independent probabilities $P(t_1)$ and $P(t_2)$ and thus, $PMI(t_1, t_2)$ will be greater than 0. Positive PMI (PPMI) is obtained by setting all negative PMI values to 0. The probability of a term t can be calculated as the document frequency of t normalised by the frequency of all words in all documents.

$$P(t) = \frac{df(t)}{\sum_{i=1}^{N} df(t_i)} \tag{2.14}$$

Where $df(t)$ returns the document frequency of t, and N is the total number of terms in the vocabulary. PMI values do not lie within the range 0 to 1 and thus, we need to introduce a normalisation operation. We normalise PMI as shown in equation 2.15.

$$Rel_{PMI}(t_1, t_2) = \frac{PPMI(t_1, t_2)}{-log_2 P(t_1, t_2)} \tag{2.15}$$

The different approaches reviewed in this section present an opportunity for studying the performance of different semantic relatedness methods for semantic indexing. However, note that all of the semantic relatedness metrics reviewed in this section, and all of the semantic indexing approaches reviewed in section 2.2 are unsupervised. This means that the semantic document representations produced are not optimised for text classification. The result of this is that the benefit of traditional semantic indexing to text classification may be limited by the lack of supervision. In the next section, we review approaches that have been proposed for supervised semantic indexing.

2.2 Semantic Indexing

Once we have understood how to compute semantic relatedness, we now need to understand how this can be utilised to address the problem of variation in indexing vocabulary. Semantic indexing is a technique that utilises semantic relatedness between vocabulary terms in order to improve document representations (Deerwester et al. 1990). Thus, given a document d represented by the

vector $\vec{d}$, the general aim of semantic indexing is to apply a transformation function on $\vec{d}$ in order to obtain a new representation $\vec{d'}$ which better models the semantic relatedness between the terms present in d and all related terms in the indexing vocabulary V as presented in equation 2.16.

$$d' = \phi(d) \tag{2.16}$$

In this way, semantic indexing aims to allow reasoning beyond the exact terms present in documents to other semantically related terms in order to improve classification performance. The effect of semantic indexing is that the representations of similar documents are brought closer together in the term-document space.

In the following sub-sections, we review a number of approaches to semantic indexing using both knowledge-resource-based (WordNet), and distributional approaches.

2.2.1 Semantic Indexing using WordNet

Early work on semantic indexing using WordNet for text classification can be found in (Scott & Matwin 1998). The proposed approach represents documents using semantic vectors of WordNet synsets. The idea is to map individual terms in a given document to synsets and their hypernyms (parents), and then using this using this to form a new vector representation for the document called a hypernym density representation. Creating the hypernym density representation involves three steps. Firstly, part-of-speech tagging is applied to all terms in the document. Secondly, the synsets and corresponding hypernyms of all nouns and verbs are obtained from WordNet. A parameter $h >= 0$ is used to limit the height of the hypernym hierarchy being considered. Lastly, the density of each synset, which is the number of occurrences of the synset in the document divided by the number of words in the document, is computed. Evaluation was performed on six text classification corpora using a rule-learning algorithm called RIPPER (Cohen 1995). Results show the hypernym density representation to lead to a reduction in error rate on two datasets compared to a standard BOW representation. However, on the remaining four datasets, no significant improvement is observed over the BOW representation.

A more comprehensive evaluation of the hypernym density representation on larger dataset sizes is presented in (Scott 1998). Evaluation was again performed using the RIPPER learning algorithm. Results again show no significant improvement from the hypernym density representa-

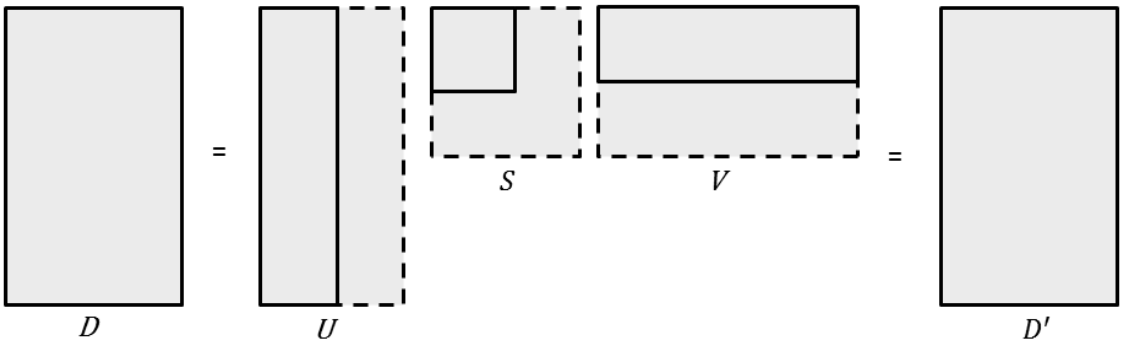

Figure 2.4: Details of LSI showing the truncation of the U, S and V matrices and the reconstructed semantic term document matrix D'

tion compared to BOW. Note that both (Scott & Matwin 1998) and (Scott 1998) do not explicitly utilise WordNet for semantic relatedness computation. Rather, WordNet synsets (concepts) are used directly for indexing. Similar approaches have also been presented in (Gonzalo, Verdejo, Chugur & Cigarrin 1998, Gomez et al. 2004, Rosso, Molina, Pla, Jimenez & Vidal 2004). In this thesis, we refer to this as concept-based (semantinc) indexing which we discuss in further detail in Section 2.5.

2.2.2 Latent Semantic Indexing

A popular semantic indexing approach is Latent Semantic Indexing (LSI) which uses singular-value decomposition (SVD) to exploit co-occurrence patterns of terms in documents to create a semantic concept space which reflects the major associative patterns in the corpus (Deerwester et al. 1990). In this way, LSI brings out the underlying latent semantic structure in texts. Given a term-document matrix D, SVD is used to decompose D into three matrices: U, a term by dimension matrix; S a diagonal matrix of singular values; and V, a document by dimension matrix. By ordering the singular values in S in decreasing order of size, S can be truncated to retain only the top k largest singular values which correspond to the k most important concepts in the term-document space. The U and V matrices are also truncated to the same rank as S. The product of the rank reduced U, S and V matrices produces a term-document matrix D' where the latent semantic structure of documents and terms are better modelled. This process is illustrated in Figure 2.4.

Figure 2.5 illustrates the process of semantic indexing of a document collection using LSI. Ini-

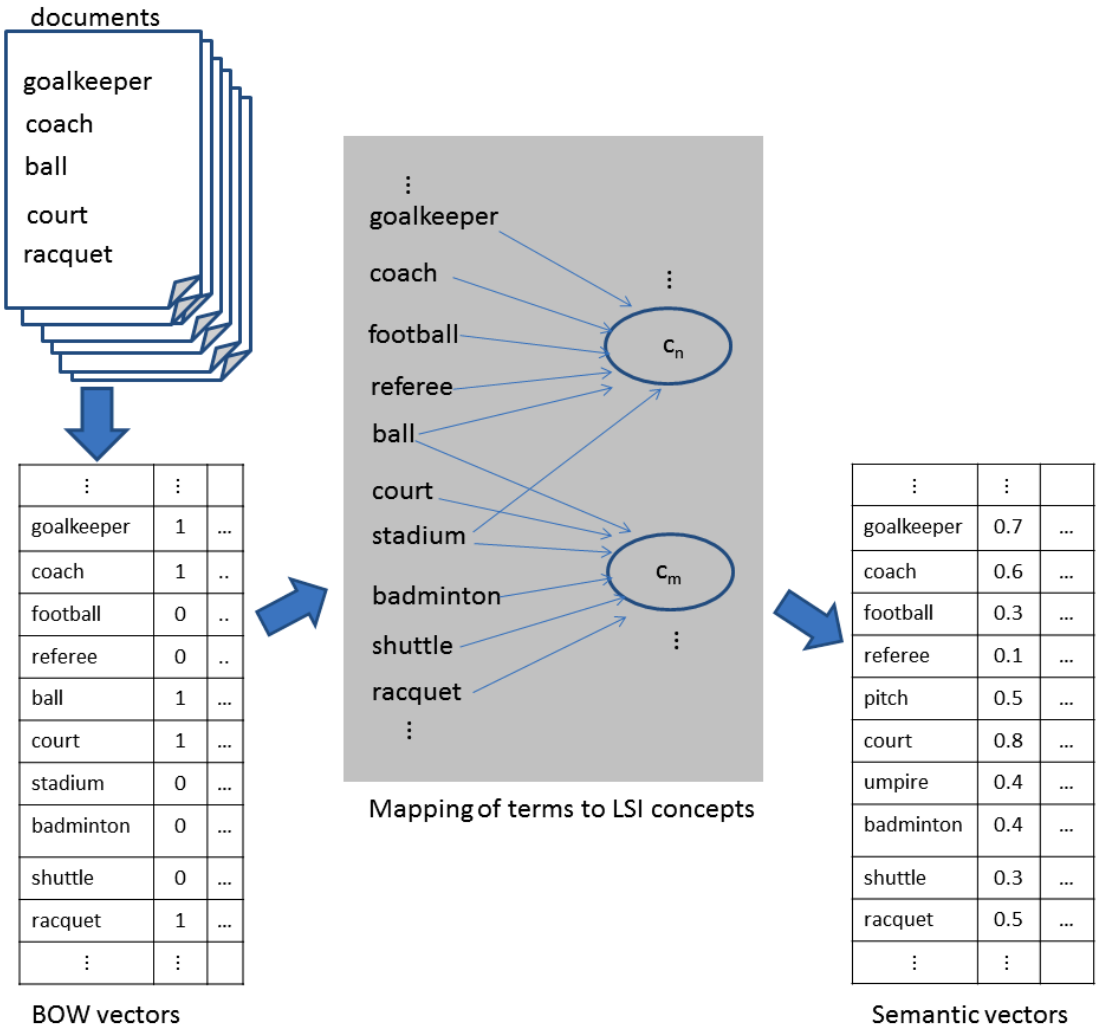

Figure 2.5: Semantic indexing of a document using LSI

tial BOW vector representations are generated for the documents in the collection which together form the term-document matrix. LSI is then applied on the entire term-document matrix in order to map individual terms in the indexing vocabulary to latent concepts. The result of this process is that document representations are transformed such that the weights of terms in the new semantic vectors produced, reflect the membership of these terms to the latent concepts in the collection. This way, the sparseness in the document vectors is reduced because the conceptual relatedness between terms is now captured in the new document representations

LSI has a number of limitations as a framework for semantic indexing. Firstly, LSI provides little flexibility over how the semantic relatedness of terms is computed. Semantic relatedness

between terms are implicitly computed in the SVD process and captured in the resulting rank-reduced term-document matrix D'. However, what if one wishes to use another approach for computing semantic relatedness and not SVD? It is not clear from figure 2.1.2 how one can introduce semantic relatedness computed using other approaches for use in semantic indexing.

Secondly, many instances of poor text classification performance from semantic indexing using LSI have been reported in the literature. For example LSI was found to produce very poor results for text classification on the 20 Newsgroup dataset. (Zelikovitz & Hirsh 2001). Similarly, LSI was also found to perform poorly in text classification on the Reuters 21578 dataset, compared to standard non-semantic VSM representation (Zhang, Yoshida & Tang 2008). An extensive evaluation of LSI on several text classification datasets using both kNN and SVM classifiers is presented in (Cachopo 2007) and on the 20 Newsgroup dataset, LSI was found to consistently lead to a decrease in classification accuracy. An explanation for this poor performance of LSI is provided in (Zelikovitz & Hirsh 2001) and (Liu, Chen, Zhang, ying Ma & Wu 2004) where the authors attribute the poor performance of LSI to its inability to capture the discriminatory characteristics of the respective classes in document representations.

A third limitation of LSI is computational cost. The SVD matrix decomposition is a computationally expensive operation. In most cases, computing SVD for large document collections is impractical and LSI is typically applied only to a sampled subset of documents instead of the entire collection (Schütze, Hull & Pedersen 1995).

These limitations of LSI highlight the need for a framework that explicitly separates between semantic indexing and semantic relatedness computation. Indeed, many approaches have been proposed for computing semantic relatedness (some of these approaches are discussed in Section 2.1) and it is important to review the performance of semantic indexing using these individual approaches on text classification. To achieve this, a flexible framework is required that allows semantic relatedness computed using any approach to be utilised for semantic indexing.

2.2.3 Latent Dirichlet Allocation

Another semantic indexing approach worth mentioning is Latent Dirichlet Allocation (LDA). LDA is a generative probabilistic model in which each term in the vocabulary is modelled as a finite mixture over a set of topics, and each topic is modelled as a mixture over a set of topic probabilities. One of the goals of LDA is to find more concise descriptions of documents in a collection while

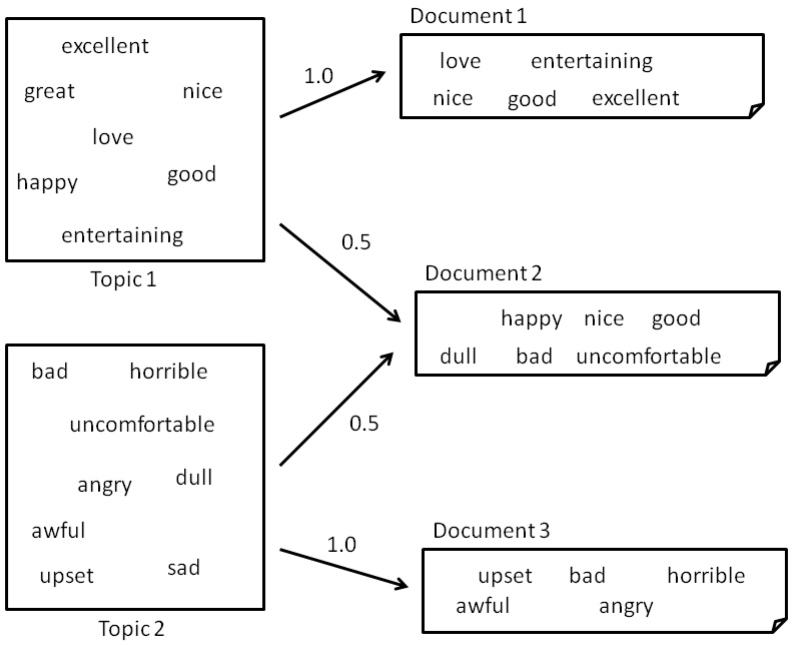

Figure 2.6: Document generation using LDA.

preserving the essential relationships that are useful for estimating similarity between documents. Accordingly, LDA aims to discover useful statistical relations between terms in a corpus. It is assumed that the resulting topics are a more accurate estimation of the semantics of documents. Thus, by reducing document representations to the space of latent topics from the corpus, much of the inherent semantic relatedness between the individual terms in the vocabulary are implicitly captured in the new representation. LDA is based on the intuition that when writing a document, the author typically thinks of a number of topics that are relevant to that document with different probabilities of relevance. The author then proceeds to draw terms from these topics in order to compose the document as show in figure 2.6. Thus, given any document d with observed words *w*, the relevant topic distribution can be obtained by inferring the probability distribution of the words *w* over all topics (Steyvers & Griffiths 2007).

For text classification, we need to be able to compute document similarity using LDA-based representations. The similarity between any two documents d_1 and d_2 can be computed by measuring the similarity of their corresponding topic distributions θ_1 and θ_2. The distribution of topics over a document can be regarded as a feature vector of the document in the space of topics. Thus, similarity between documents can be computed using any standard geometric similarity functions

e.g. cosine as shown in Equation 2.17.

$$Sim(d_1, d_2) = \frac{\sum_i^k \theta_{1,i}\theta_{2,i}}{\parallel \theta_1 \parallel \parallel \theta_2 \parallel} \tag{2.17}$$

Unlike LSI, LDA is not designed particularly for the VSM even though, as equation 2.17 shows, it can be adapted for use in the VSM. For this reason, LDA is more widely used in probabilistic text retrieval models which are a more natural fit for the LDA model (Wei & Croft 2006). Also, similar to LSI, LDA provides very little flexibility over how semantic relatedness is computed. Rather, semantic relatedness between terms is inherently captured in the probabilistic mapping of terms to latent topics. This further highlights the advantage of having a framework that separates between semantic indexing and semantic relatedness computation.

2.2.4 Generalised Vector Space Model

The Generalised Vector Space Model (GVSM) was introduced in (Wong et al. 1987) as a technique for introducing a measure of relatedness between terms into document vector representations. In the GVSM, all terms t_i in the indexing vocabulary V are assumed to have a corresponding vector representation $\vec{t_i}$ in euclidean space. Accordingly, the relatedness between any two terms t_i and t_j can be computed as a function of the distance between their vector representations $\vec{t_i}$ and $\vec{t_j}$. The relatedness between any two terms is thus represented by a numerical value where totally unrelated terms have a relatedness value of zero and higher values represent stronger relatedness. If we assume (for sake of simplicity) that the similarity between any two documents q and d is obtained as the dot product of their respective vector representations $\vec{q}$ and $\vec{d}$, then this can be obtained in the GVSM as shown in equation 2.19:

$$Sim(\vec{q}, \vec{d}) = \sum_i^{|q|} \sum_j^{|d|} w_i \vec{t_i} w_j \vec{t_j} \tag{2.18}$$

$$Sim(\vec{q}, \vec{d}) = \sum_i^{|q|} \sum_j^{|d|} w_i w_j \vec{t_i} \vec{t_j} \tag{2.19}$$

Where w_i and w_j are the initial (*tf-idf*, binary e.t.c.) weights for the terms $t_i \in d$ and $t_j \in q$ respectively. Note that the product of the two term vectors, $\vec{t_i}$ and $\vec{t_j}$, provides the semantic

relatedness between the corresponding terms t_i and t_j as shown in equation 2.20.

$$Rel(t_i, t_j) = \vec{t_i}\vec{t_j} \tag{2.20}$$

Therefore, the term vectors $\vec{t_i}$ and $\vec{t_j}$ need not be known so long as the similarity between terms t_i and t_j ($rel(t_i, t_j)$) is known (Tsatsaronis & Panagiotopoulou 2009). Accordingly equation 2.19 can be rewritten as follows:

$$Sim(\vec{q}, \vec{d}) = \sum_{i}^{|q|} \sum_{j}^{|d|} w_i w_j rel(t_i, t_j) \tag{2.21}$$

Thus, equation 2.21 allows any approach to be used for obtaining $rel(t_i, t_j)$. This way, the GVSM provides a convenient framework where the computation of semantic relatedness ($rel(t_i, t_j)$) is separated from semantic indexing. This allows any effective approach for the computation of semantic relatedness to be utilised for semantic indexing.

Semantic indexing using the GVSM model has been widely applied to text classification albeit sometimes without explicit reference to the name GVSM e.g. (Chakraborti, Wiratunga, Lothian & Watt 2007, Gabrilovich & Markovitch 2009, Nasir, Karim, Tsatsaronis & Varlamis 2011). Note also that LSI can be used with the GVSM where SVD is used for acquiring semantic relatedness between terms and GVSM is used for semantic indexing. This further demonstrates the advantage of separating semantic relatedness computation from semantic indexing. In the next sub-section, we present a detailed review of several approaches that have been proposed for semantic relatedness computation .

2.3 Supervised Semantic Indexing

The main limitation of conventional semantic indexing approaches for supervised tasks is that these techniques are agnostic to class knowledge. This means that the semantic representations produced using these approaches are not necessarily the best fit for the class distribution of the document collection (Aggarwal & Zhai 2012). This is a well recognised problem and a number of supervised extensions to traditional semantic indexing approaches have been proposed. We discuss the most popular of these approaches in the following sub sections.

2.3.1 Supervised LSI

An extension of LSI called supervised LSI (SLSI) that iteratively computes SVD on term similarity matrices of separate class is presented in (Sun, Chen, Zeng, Lu, Shi & Ma 2004). A separate term-doc matrix is constructed for each class and in each iteration, SVD is performed on each class-specific term-doc matrix. The most discriminative eigen vector across all categories is selected as the basis vector in the current iteration. The effect of the selected eigen vector is then subtracted from the original term-document matrix. The iteration continues until the dimension of the resulting space reaches a predefined threshold. The evaluation compared three types of representations: standard BOW without semantic indexing, unsupervised LSI and SLSI using kNN and SVM classifiers. Results show SLSI performs better than LSI. However, SLSI only achieved marginal gains over BOW using kNN while both SLSI and LSI failed to perform better than SVM.

2.3.2 Sprinkled LSI

A more promising supervised extension to LSI which uses an approach called sprinkling where class-specific artificial terms are appended to representations of documents of the corresponding class (Chakraborti, Lothian, Wiratunga & Watt 2006). LSI is then applied on the sprinkled term-document space resulting in a concept space that better reflects the underlying class distribution of documents. An overview of the sprinkling process is shown in Figure 2.7.

Sprinkling involves generating a set of artificial terms for each class in the training corpus. Document representations is the term-document matrix D are then augmented with the artificial terms that correspond to their respective class. A higher order term-relatedness approach e.g. LSI is then applied on the augmented term-document space which results in stronger associations between terms that occur more often within documents of the same class. An important consideration for sprinkling is the number of artificial terms to sprinkle. In (Chakraborti et al. 2006), the authors found sprinkling 16 terms per-class to give optimal performance. A more sophisticated approach called adaptive sprinkling which optimises the number of sprinkled terms for each individual dataset based on dataset complexity is presented in (Chakraborti, Wiratunga, Lothian & Watt 2007). Adaptive sprinkling exploits the confusion matrix of each dataset produced by a classifier. A confusion matrix records the performance of the classifier such that the columns of the matrix represent the instances predicted by the classifier and the rows represent the actual instances

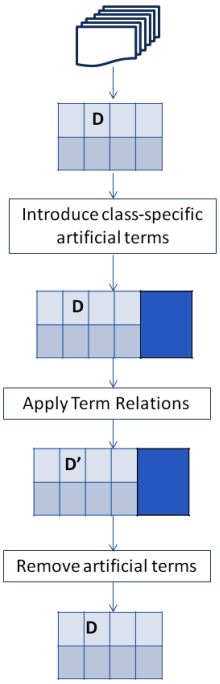

Figure 2.7: Sprinkling.

that belong to the class. The non-diagonal entries of the confusion matrix therefore represent the instances the are misclassified by the classifier. The larger the entry in a non-diagonal cell, the harder that class is to the classifier. In this way, adaptive sprinkling allocates more artificial terms to the harder classes.

Sprinkled LSI was compared with unsupervised LSI and SVM on a number of classification tasks. Results showed sprinkled LSI to significantly out perform both unsupervised LSI and SVM. However, a major limitation of sprinkling and adaptive sprinkling is that both techniques are only applicable to higher order term relations. This is because the 'sprinkled' term-document space has no effect on first-order term relations. Therefore, there is a need for a more general approach for utilising class knowledge for semantic indexing. Particularly, we need a method that is independent of the type and order of semantic relatedness. Furthermore, adaptive sprinkling requires the number of artificial terms used for sprinkling to be optimised for each individual class which introduces a significant overhead if the number of classes is large.

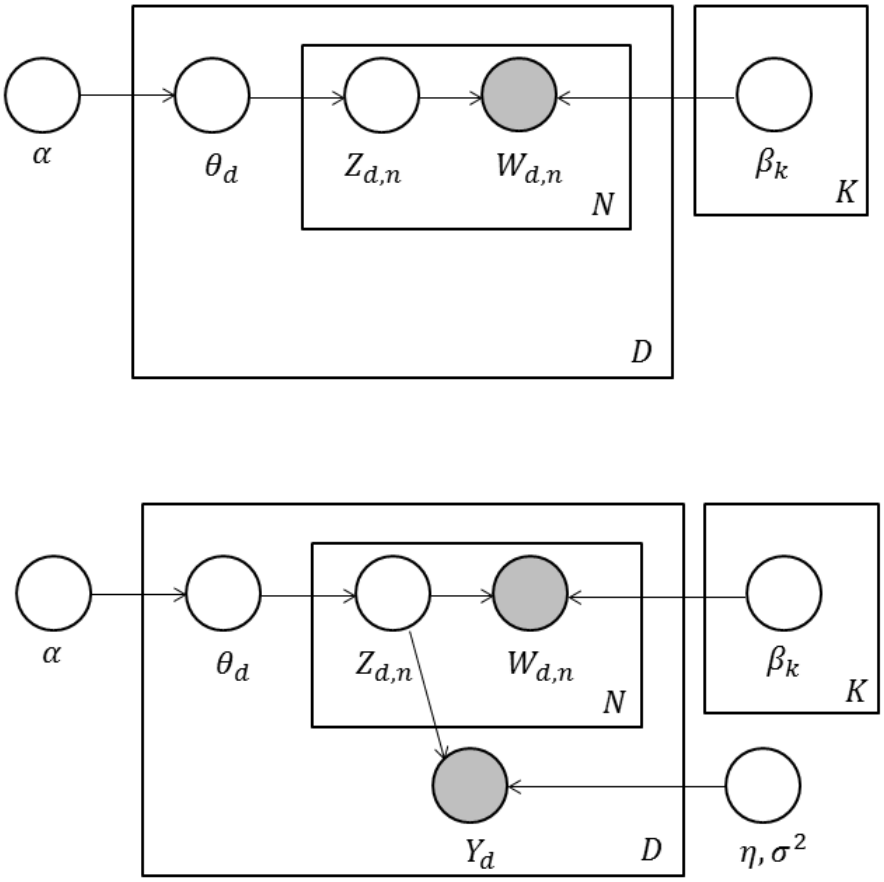

Figure 2.8: Graphical model of LDA and SLDA

2.3.3 Supervised LDA

A supervised version of LDA called sLDA is presented in (Blei & McAuliffe 2008). Here, a response variable (class label, real value, cardinal or ordinal integer value) associated with each document is added to the LDA model. Thus the topic model is learned jointly for the documents and responses such that the resultant topics are good predictors of the response variables. The difference between LDA and sLDA topic modelling approaches is illustrated in figure 2.8.

Figure 2.8 is a graphical model representation of LDA (top) and sLDA (bottom). As can be observed, the main difference between the two models is that sLDA includes a response variable Y_d which is conditioned on the response parameters η and δ. This means that prediction is also built into sLDA i.e. given any document, it is possible to predict the response variable Y_d directly from the sLDA model without the need to use any classifier. Thus, sLDA is more than simply a semantic indexing technique.

The predictive performance of sLDA on two regression tasks compared with LDA and lasso (L_1-regularized linear regression) suggests moderate improvements on both tasks. However, sLDA inherits much of the disadvantages of LDA including computational cost and choice of an optimal number of topics.

2.4 Supervised Document Indexing

Term weighting is a critical part of document indexing in the VSM. The goal of term weighting is to assign, for each term t_j in the indexing vocabulary and for each document d_i, a weight $w_{i,j}$ which represents how much t_j contributes to the discriminative semantics of d_i. Because of the unsupervised nature of the traditional *tf-idf* term weighting scheme, it is not likely to be optimal for text classification. In particular, the suitability of *idf* for text classification has been challenged (Debole & Sebastiani 2003). The aim of *idf* is to assign higher weight to terms that better distinguish the small set of documents that are likely to be relevant to any given query from the much larger set of irrelevant documents in the collection. Note that this assumption is more intuitive for information retrieval where typically, a large heterogeneous collection of documents is expected to cater for a diverse multitude of user information needs or topics. However, for text classification, the set of topics (i.e. classes) are much fewer (in many cases just two) and are explicitly labelled in the training collection. Thus, a number of supervised document indexing approaches have proposed replacing the *idf* component of the *tf-idf* weighting scheme with a supervised alternative which better captures the class distribution of terms as presented in equation 2.22 (Debole & Sebastiani 2003, Deng, Tang, Yang, Li & Xie 2004, Lan, Tan & Low 2006).

$$w_{i,j} = tf_{i,j} \times \delta(t_j) \tag{2.22}$$

Where $w_{i,j}$ is the weight of term t_j in document d_i, $tf_{i,j}$ is the term frequency of t_j in d_i and $\delta(t_j)$ is a function that returns the supervised weight of t_j. In practice, $\delta(t_j)$ is typically obtained using supervised feature selection metrics e.g. Chi-square, Information Gain, Gain Ratio or Mutual Information. For example, supervised weighting with χ^2 using the approach presented in equation 2.22 is as shown in equation 2.23.

$$w_{i,j} = tf_{i,j} \times \chi^2(t_j) \tag{2.23}$$

Given the entire vocabulary V of a document collection, feature selection is a technique used for selecting a subset $U \subset V$ of the most important terms for use as an optimised indexing vocabulary. This involves computing for each term $t_j \in V$, a statistical score of term importance which is used to rank all terms in V. Terms that rank below a certain threshold are subsequently excluded from the new indexing vocabulary U. Note that this score of term importance can be used as a weight for terms such that more important terms have a greater contribution to document representation.

Feature selection approaches can be categorised into supervised and unsupervised. For the purpose of this discussion, we will focus exclusively on supervised feature selection metrics. Many supervised feature selection techniques have been proposed in the literature which include Information Gain (IG), Chi squared (χ^2), Mutual Information, Gain Ratio (GR) and Odds Ratio (OR). The mathematical formulations of these feature selection metrics are given in table 2.1, where $p(t_j, c_k)$ is the probability that a document contains the term t_j and belongs to the class c_k, $p(t_j)$ is the probability that a document contains the term t_j, and c_k is the probability that a document belongs to class c_k.

Function	Formula
Chi-squared	$\chi^2(t_j, c_k) = \frac{(P(t_j,c_k)P(\bar{t_j},\bar{c_k}) - P(t_j,\bar{c_k})P(\bar{t_j},c_k))^2}{P(t_j)P(\bar{t_j})P(c_k)P(\bar{c_k})}$
Information Gain	$IG(t_j, c_k) = \sum_{c \in \{c_k, \bar{c_k}\}} \sum_{t \in \{t_j, \bar{t_j}\}} P(t,c) log \frac{P(t,c)}{P(t)P(c)}$
Gain Ratio	$GR(t_j, c_k) = \frac{\sum_{c \in \{c_k, \bar{c_k}\}} \sum_{t \in \{t_j, \bar{t_j}\}} P(t,c) log \frac{p(t,c)}{p(t)p(c)}}{\sum_{c \in \{c_k, \bar{c_k}\}} P(c) log P(c)}$
Mutual Information	$MI(t_j, c_k) = log \frac{P(t_j,c_k)}{P(t_j)P(c_k)}$
Odds Ratio	$OR = \frac{df(t_j,c_k)/df(\bar{t_j},c_k)}{df(t_j,\bar{c_k})/df(\bar{t_j},\bar{c_k})}$

Table 2.1: Supervised Feature selection metrics.

Let N be the number of documents in the collection, N_{c_k} the total number of documents that belong to class c_k, $df(t_j)$ the number of documents in the collection that contain the term t_j, $df(t_j, c_k)$ the number of documents belonging to class c_k that contain term t_j. Then, the probabilities in table 2.1 can be computed as follows:

$$p(t_j, c_k) = df(t_j, c_k)/N$$
$$p(t_j) = df(t_j)/N$$
$$p(c_k) = N_{c_k}/N$$

A comparative analysis on feature selection techniques for text classification found χ^2 and IG to give the best performance (Yang & Pedersen 1997). These results are supported by another comparative study of a larger set of feature selection metrics where IG and χ^2 were found to give the best performance in terms of precision (Forman 2003). IG is a measure of the information available for category prediction by knowledge of the presence or absence of a term in class. Thus, the higher IG value of a term t_j, the more important t_j is for class prediction. On the other hand, χ^2 measures the lack of independence between a term t_j and a class c_k. Accordingly, the higher the χ^2 score of a term t_j the more important t_j is for class prediction. Despite the differences in the fundamental approach of IG and χ^2 to feature selection, both techniques are good at measuring the predictiveness of terms, hence their good performance on feature selection. This means that both IG and χ^2 are likely to produce good results when used for providing supervised term weights. Indeed, this intuition is supported by results of a comparative study of *tf-idf* and supervised term weighting approaches presented in (Deng et al. 2004). The supervised weights considered are: *tf-CHI* which combines *tf* with χ^2, and *tf-OddsRatio* which combines *tf* with Odds Ratio where, in both cases, tf is combined with the supervised weight as shown in equation 2.22. Results of a comparative evaluation on text classification using SVM showed *tf-CHI* to outperform the other weighting schemes while the second best weighting scheme was *tf-OddsRatio*.

A more extensive comparative analysis using different classifiers: Rocchio, SVM and KNN, is presented in (Debole & Sebastiani 2003). Here also, the authors use the same approach as equation 2.22 for supervised term weighting using χ^2, Gain Ratio (GR) and Information Gain (IG), and compared these with standard *tf-idf*. Of the three supervised weighting approaches, GR produced the best result across all three classifiers followed by χ^2. Supervised weighting with IG was found to produce rather disappointing results. Also, the results show that supervised weighting does not always produce improvements as all three supervised approaches were outperformed by *tf-idf* on a number of datasets.

Mixed performance from supervised weighting was also reported in (Lan et al. 2006). Here, the authors propose a new supervised weighting component called *relevance factor* (rf) which assigns a weight $w_{i,j}$ to a given term t_i with respect to class c_j, proportional to the relative frequency t_i in $c_j$as shown in equation 2.24.

$$rf = log(2 + \frac{f(t_i, c_j)}{f(t_i, \bar{c_j})}) \tag{2.24}$$

Where $f(t_i, c_j)$ is the frequency of t_i in c_j and $f(t_i, \bar{c_j})$. A comparative evaluation was performed to compare supervised term weighting using *tf-rf*, *tf-CHI*, *tf-IG* and *tf-OddsRatio*, and unsupervised term weighting using term frequency (*tf*), *tf-idf* and binary on text classification using kNN and SVM. Supervised weighting was done using equation 2.22. Results showed that supervised term weighting was not consistently better than unsupervised term weighting. Of the supervised term weighting approaches, only *tf-rf* was found to outperform *tf-idf*. The other supervised term weighting approaches, *tf-CHI*, *tf-IG* and *tf-OddsRatio*, all performed consistently worse than *tf-idf*.

The lack of consistent improvement from supervised document indexing indicates that effective use of supervised weighting for document representation remains an open research problem. Indeed, all the approaches reviewed share the same assumption that that the *idf* component of *tf-idf* should be replaced by a supervised weighting scheme e.g. χ^2 or IG. However, while the intuition for the introduction of supervised weights is sound, the need to replace *idf* is less so. Indeed, empirical evidence shows that *idf* does work well for text classification (Debole & Sebastiani 2003, Lan et al. 2006). Thus, given the success of *tf-idf* in text classification, there is no sound justification why a supervised weighting scheme needs to replace *idf*. An effective supervised document indexing scheme should be able to introduce supervision by building upon *tf-idf* (or any other successful term weighting scheme). This way, supervised document indexing should be able to combine the benefits of both unsupervised weighting e.g. *idf*, and supervised term weights in a systematic fashion.

2.5 Concept-Based Document Indexing

The limitation of terms to adequately model the semantics of text documents means that sometimes, more semantically rich indexing units are required. In this section, we will review a number

of proposed approaches for the indexing of documents using conceptual information. These approaches typically represent documents using vectors of concepts from a lexicon or an ontology. Most often, general purpose lexicons and ontologies are used for these approaches e.g. WordNet and The Kim Ontology (Popov, Kiryakov, Kirilov, Manov, Ognyanoff & Goranov 2004).

Several approaches have been proposed for indexing of text documents using WordNet concepts (Gonzalo et al. 1998, Scott 1998, Gomez et al. 2004, Rosso et al. 2004). The primary aim of WordNet-based approaches is to address word ambiguity by mapping terms to specific, disambiguated concepts in WordNet. Thus, these approaches are largely similar, and mainly differ on details such as the specific approach used for mapping from document terms to WordNet concepts. Other considerations include how to filter irrelevant concepts and how to assign weights to concepts in the new document representation (Gomez et al. 2004). For document indexing, these approaches either replace the Bag-Of-Words (BOW) vector with a Bag-Of-Concepts (BOC) vector (Scott 1998, Rosso et al. 2004), or augment the BOW vector by introducing new dimensions for concepts (Gomez et al. 2004). An important limitation of these WordNet-based approaches however, is that they are sensitive to the quality of word sense disambiguation used for mapping terms to concepts (Gomez et al. 2004).

Other approaches have proposed an extended VSM that uses concepts from ontologies for semantic indexing of documents (Kiryakov et al. 2004, Popov et al. 2004, Vallet, FernÃąndez & Castells 2005). These approaches also index document using either an exclusive concept-based vector (Vallet et al. 2005) or using a combination of a BOW vector and a BOC vector (Kiryakov et al. 2004, Popov et al. 2004). The advantage of using ontologies for document indexing compared to WordNet is that the set of concepts in ontologies typically contain good coverage of named entities (i.e. persons, places, organisations e.t.c.). Ontologies also contain rich semantic connections (relationships) between concepts which allow for inferential reasoning (e.g. that 'Nile' is an instance of the concept River). This makes the use of ontologies particularly attractive for use in IR because they allow for answering complex queries such as "give me a list of all rivers in Africa". However, note that achieving this requires concepts to be extracted form the both queries and text documents, and mapped to concepts in the ontology. Also, mapping extracted concepts to an ontology is only necessary if the target ontology contains relationships of interest, and also, if mechanism exist to support this type of inference.

The use of ontologies for document indexing presents interesting opportunities. However,

ontology-based document indexing is a research area that is still in its infancy and requires satisfactory levels of performance at many stages (e.g. having a suitable ontology, a mapping approach from document terms to concepts, inference mechanisms e.t.c.) in order to derive benefit (Vallet et al. 2005). Accordingly, in this thesis, we investigate the use of information extraction to derive a concept-based indexing vocabulary directly from the contents of text documents. An important advantage of this approach is that our indexing vocabulary is not limited to the set of concepts available in WordNet , or any ontology.

2.6 Datasets

The evaluation of the techniques and algorithms developed in this thesis is carried out using text classification datasets covering various different domains including news stories, incident reports, medical abstracts, online reviews and discussion forums. The variety in domain of these datasets is designed to allow for a more robust evaluation of our approaches. Also, these corpora are designed for a variety of different classification tasks e.g. sentiment classification (Movie Reviews, Amazon Reviews, Twitter Dataset), topic classification (Reuters Volume 1, Ohsumed, 20 Newsgroups) and semantic classification (Incident Reports). This also allows us to evaluate the suitability of our approaches for various different types of classification tasks.

In the experiments in this thesis, we use a total of 37 binary-class datasets, and 5 multi-class datasets, each with equal number of documents in each class. Binary classification is important because most text classification problems consist of binary classification tasks (Sebastiani 2002). For example, the first group of 13 datasets come from the Ohsumed corpus. Also, multi-class classification problems can easily be framed as a number of binary classification tasks. These datasets are created from a number of different source corpora as shown in 2.2. An overview of these datasets and their corresponding source corpora is given in Table 2.2. We describe these corpora in detail in the following sub-sections. Our binary-class datasets were created by combining documents from similar classes e.g. the HardW dataset is a combination of the 2 hardware classes of the 20 Newsgroups dataset i.e. comp.sys.ibm.pc.hardware and comp.sys.mac.hardware. These types of datasets are expected to represent a more challenging classification boundary because of the similarity between the two classes. A complete listing of the combination of classes for each dataset is given in Appendix C.

Datasets	Corpus	Ave Dataset Voc. Size (Terms)	Ave. Doc. Length (Terms)
BactV, CardR, NervI, MouthJ, NeopE, DigNut, MuscS, EndoH, MaleF, PregN, ImmunoV, NervM, RespENT, Ohsumed01, Ohsumed02, Ohsumed03, Ohsumed04	Ohsumed	13,000	65
Hardw, MedSp, CryptE, ChrisM, MeastM, GunsM, AutoC, Science	20 Newsgroups	15,980	76
StratM, EntTour, EqtyB, FudA, InRelD, NProdRes, ProdNP, OilGas, ElectG	Reuters	18,304	104
Fire, Collision, Rollover, CollRoll, MiscInc, CraneFP, ShovFP	Incident Reports	1,340	19
MovieRev	MovieReviews	33,345	232

Table 2.2: Datasets used in this thesis and their source corpora, along with statistics of average vocabulary size and average document length.

2.6.1 Ohsumed

This is a subset of MEDLINE, an online database of medical literature, and comprises a collection of 50,216 medical references from medical journals from the year 1991 [2]. The Ohsumed collection is unequally divided into 23 classes according to different disease types e.g. Virus Diseases. This corpus contains documents written in clean language with a high number of domain-specific medical terms. The original categorisation of documents in this collection is non-disjoint which means the same document can be categorised under two or more different classes if it is relevant to all those classes. For our experiments, we selected only documents that belong to a single class.

We created a total of 13 binary class datasets from this corpus, each dataset containing 100 documents, balanced equally between the two classes. The 13 datasets have an average vocabulary size of about 13,000 unique terms per dataset. The average document length of the datasets is 65 unique terms. This corpus has widely been used in topic-based text classification experiments (Joachims 1998)

[2] Available for download at http://disi.unitn.it/moschitti/corpora/ohsumed-all-docs.tar.gz

2.6.2 20 Newsgroups

This corpus is a collection of 20,000 documents collected from Newsnet newsgroups messages [3]. The collection is partitioned almost equally into 20 classes of 1,000 documents each, according to newsgroup topics. For example, the class sci.space contains messages relating to space. The corpus contains documents with user generated content which means that spelling is not perfect. Documents have an email-style format which means that they often contain address headers and signature which contain information such as email addresses, names and addresses. Documents also contain replies to previous messages where the previous message is quoted in the document. All these make the 20 Newsgroups a noisy corpus where the content of documents are mixed with non-topic text.

We created 7 binary datasets from this corpus where each dataset contained 500 documents in each class. The total vocabulary size of the documents is 15,980 unique terms and the average document length is 76 terms.

2.6.3 Reuters Volume 1

This corpus is an archive of 806,791 news stories provided by the global news provider, Reuters (Lewis, Yang, Rose & Li 2004). This corpus is available by making an application to Reuters Ltd [4]. The collection comprises all news stories produced by Reuters journalists within a one year period starting from August, 1996. Documents within the collection are tagged with descriptive metadata specifying codes for topic, region and industry sector. Topic codes represent the subject area of each news story. These are organised into four hierarchical groups with top-level categories: Corporate/Industrial (CCAT), Economics (ECAT), Government/Social (GCAT) and Markets (MCAT). Industry codes are used to indicate the type of business or industry referred to by the news story and are also arranged in a hierarchy. Region codes indicate the geographical region referred to in the news story. Only topic codes and industry codes where used when creating datasets for our evaluations.

Documents in this corpus are produced by professional news journalists which means that they are often written in clean language without misspellings. However, many documents also contain

[3] Available for download at http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html

[4] Details of how to obtain this Corpus is available http://trec.nist.gov/data/reuters/reuters.html

tabular data where much of the content is numbers (e.g. share prices). These types of documents present potential challenges for identifying a clear class boundary. This is because having a high frequency of numbers in documents tends to lead to a more sparse document representations as different instances of the same number do not mean the same thing.

A total of 9 binary-class datasets were created from the Reuters corpus with an average vocabulary size of 18,304 unique terms per dataset The average document length is 104 unique terms. Two of these datasets, OilGas and ElectG, constitute of classes from the industries, CRUDE OIL EXPLORATION & NATURAL GAS EXPLORATION, and ELECTRICITY PRODUCTION & GAS PRODUCTION respectively, rather than topic.

2.6.4 Incident Reports

This corpus was created using incident reports crawled from the Government of Western Australia's Department of Mines and Petroleum website [5] in November 2011. Incident reports are organised on the website in categories e.g. **Outbreak of fire**, indicating the nature of the incident. The distribution of reports in each category is quite variable with some categories having less than 50 reports and others having more than 200. We selected only categories having more than 200 reports.

Under each incident category, incident reports are further classified into **Injury** and **NoInjury** categories depending on whether or not injuries were sustained in the incidents they describe. At the time of crawling the website, each incident report was available as a single html file which we downloaded and extracted the incident description from, in order to create the datasets. Incident reports are professionally written and clean. However, they are also very brief and straight to the point in their description.

We created a total of 7 datasets with an average vocabulary size of 1,340 unique terms per dataset. Documents in the datasets have an average length of 19 terms. Each dataset contains a total of 200 documents distributed equally over the two classes (i.e. 100 documents per class). The datasets have been made available online [6].

[5] http://dmp.wa.gov.au

[6] https://www.dropbox.com/sh/myrdhqq9ccf00dd/AABIBmfZhTzRypdCWum7oBF-a?dl=0

2.6.5 Movie Reviews

This is a sentiment classification corpus comprising movie reviews from the Internet Movie Database (IMDB) (Pang, Lee & Vaithyanathan 2002). We used version 1.0 of this corpus which contains 1400 reviews, half of which are classified as expressing positive sentiment while the other half is classified as negative [7]. Accordingly, the classification task for this dataset is to determine the sentiment orientation of any given review.

Despite this corpus being popularly referred to as a movie reviews dataset, documents contain much more than just the review of the movie including a list of cast and a synopsis. We treat the entire corpus as a single dataset. This dataset has a vocabulary size of 33,345 unique terms and an average document length of 232 terms, making this the dataset with the longest documents in our collection.

2.7 Chapter Summary

In this chapter, we discussed text representation using the Vector Space Model (VSM). We showed how the use of the traditional VSM for text classification suffers three major limitations. The first is the problem of variation in indexing vocabulary. This is commonly addressed using semantic indexing approaches which aim to capture semantic relation between terms and use this information to generalise document representations away from low-level expressions to higher-level semantic concepts. Much work has been done in using semantic indexing for text classification. However, the lack of consistent improvement indicates that a proper investigation into the role of semantic indexing for text classification is required. Accordingly, in this thesis, we evaluate the performance of semantic indexing on text classification tasks. This evaluation allows us to answer one important question, how beneficial is semantic indexing for text classification? We also carry out a detailed analysis of the semantic indexing process in order to identify reasons why semantic indexing may lead to poor text classification performance. Based on our findings, we propose a semantic indexing framework that addresses the limitations identified in our analysis.

The second limitation of the VSM is the problem of suboptimal document vectors due to the lack of supervision. We reviewed a number of approaches that have been proposed for supervised

[7] Download at http://www.cs.cornell.edu/people/pabo/movie-review-data/mix20_rand700_tokens_cleaned.zip

document indexing using supervised term weighting. These approaches use the popular tf-$\delta(t)$ technique which combines term frequency (*tf*) with a supervised weighting function ($\delta(t)$). However, the lack of conclusive improvements from these supervised indexing approaches indicates that the proper use of supervised weighting for document representation remains an open research challenge. Indeed, all the approaches reviewed share the same assumption that the *idf* component of *tf-idf* should be replaced by a supervised weighting scheme e.g. χ^2 or IG. However, the need to replace *idf* is not well motivated especially when empirical evidence shows that *idf* does work well for text classification (Debole & Sebastiani 2003, Lan et al. 2006). Thus a proper framework for supervised document indexing should be able to combine the advantages of both *idf* and supervised term weights in a systematic fashion. Accordingly in this thesis, we investigate the application of our indexing framework for the supervised document indexing.

Traditional semantic indexing approaches are not optimised for text classification because of the lack of supervision at all stages of the process. We reviewed a number of approaches that have been proposed for supervised semantic indexing e.g. supervised LSI (SLSI), sprinkled LSI (SprLSI) and supervised LDA (SLDA). However, evaluation of SLSI has not shown conclusive improvement over LSI while SprLSI requires complex parameter tuning e.g. the optimal number of terms to use for sprinkling which so far needs to be determined individually for each dataset. SLDA also requires complex parameter tuning e.g. the optimal the number of topics to use. Furthermore, sLDA is a computationally expensive process and can easily take several hours to complete even on small datasets (Xu, Chen, Weinberger & Sha 2012). These reasons make simpler and computationally efficient semantic indexing approaches preferable.

The limitation of a term-based indexing vocabulary is typically addressed using either an indexing vocabulary of concepts from a lexicon e.g. WordNet or an ontology. However, the two options are far from being satisfactory solutions. For certain domains and tasks e.g. semantic indexing of incident reports, both approaches do not capture the type of semantic concepts (i.e. events) needed for effective document indexing. Accordingly, in Chapter 7 of this thesis, we investigate the use of information extraction for semantic indexing of incident reports.

Chapter 3

When to use Semantic Indexing

Semantic indexing is used to address the problem of variation in indexing vocabulary by discovering semantic relations between terms and using this knowledge to identify conceptual similarity. The expectation is that the semantic representations produced by semantic indexing should lead to better text classification performance. Indeed, semantic indexing using LSI, as well as semantic relatedness mined using first and higher order term associations were found to improve text classification performance in (Chakraborti, Wiratunga, Lothian & Watt 2007). An evaluation of LSI on six classification datasets also showed the average performance of LSI to be better than that of a basic BOW representation (Cardoso-cachopo, Tulisbon, Av & Pais 2007).

However, although semantic representations have proven quite beneficial, it remains to be determined whether semantic indexing consistently improves text classification performance. For example, semantic indexing using LSI in (Cristianini, Shawe-Taylor & Lodhi 2002) produced no significant improvement while in (Zelikovitz & Hirsh 2001), (Liu et al. 2004) and (Zhang et al. 2008), LSI performed worse than not using semantic indexing. In (Smeaton 1997), the authors report poor document retrieval performance in an IR task, from semantic indexing using WordNet. In addition, (Basili, Cammisa & Moschitti 2005) found that semantic indexing using WordNet only improved text classification performance when limited training documents are available (in general less than 100 documents). This led the authors to conclude that semantic indexing does not improve classification performance if there is sufficient training data.

For these reasons, in this chapter we address two important questions:

1. How much improvement on text classification performance can we achieve with semantic

indexing?

2. Can we predict instances when semantic indexing is likely to improve classification performance?

We address the first question by empirically evaluating the performance of both knowledge-resource-based and distributional semantic relatedness techniques on a number of text classification datasets. To address the second question, we investigate the use of meta learning for predicting when to use semantic indexing. The objective of meta-learning is to produce proper guidance on the right algorithm to use, from a number of available algorithms and techniques, according to the nature of the problem. (Vilalta, Giraud-Carrier, Brazdil & Soares 2004). Our hypothesis is that datasets which do not benefit from semantic indexing will likely have similar attributes. Accordingly, we present several attributes of text datasets that are predictive of the performance of semantic indexing. We use these attributes in a meta case-based system to predict, given any text dataset, whether or not to apply semantic indexing for representation. Being able to accurately predict when semantic indexing is not likely to improve retrieval performance means that we can conveniently avoid the overhead of semantic indexing in the first place.

The rest of this chapter is structured as follows, in Section 3.1 we review the performance of semantic indexing on text classification tasks, using four different knowledge-resource based and three different distributional approaches for computing semantic relatedness. In Section 3.2, we introduce meta-learning and present our case-based approach for predicting when to use semantic indexing along with a description of the dataset attributes which we use for case representation. An evaluation of our case-based approach is also presented. We conclude this chapter with a summary in Section 3.3.

3.1 Performance of Semantic Indexing

We address our first question of whether semantic indexing always improves text classification performance by running a number of text classification experiments where, for each evaluation, we have two types of representations of the same dataset, one a baseline Bag-Of-Words (BOW) representation and a second semantic representation produced using semantic indexing. Accordingly, the benefit from semantic indexing can be quantified as the extent to which text classifica-

tion performance on the semantic representations improve on the BOW baseline representations. Since our representation is based on the Vector Space Model (VSM), we need a semantic indexing approach for introducing semantic relatedness into the vector representations of documents. To achieve this, we use the generalised vector space model (GVSM) which we described in Section 2.2.4. This allows us to experiment with several different approaches for computing semantic relatedness. Accordingly, our evaluation is divided into 2 subsections. In sub-section 3.1.2, we present a comparative analysis of semantic indexing using the knowledge resource WordNet for providing semantic relatedness. In subsection 3.1.3, we present semantic indexing using distributional approaches. In all cases, pairwise semantic relatedness values are computed using the respective semantic relatedness approach and provided to the GVSM for semantic indexing.

3.1.1 Experiment Setup

In all experiments in this thesis, we report classification accuracy in percentage over 5 runs of 10-fold cross validation. This means that each dataset is divided into 10 equal parts called folds, with each fold containing equal number of documents from all classes. Each fold is then used in turn as a test set and the remaining 9 parts are used for training. The 5 runs are achieved by randomly re-arranging the order of documents in each dataset and running another 10-fold cross validation. Classification is performed using a similarity weighted kNN approach with k=3, where each test document is taken in turn and the cosine similarity metric is used to identify the 3 most similar documents in the training set and the class of the documents with the highest weight (similarity) is assigned as the class of the test document. The value of $k = 3$ was chosen after conducting experiments with different values of k (3, 5, 10, 15 and 20) and no significant difference (using an Anova test) was found for the results produced by the different values of k. The value of 3 for k was chosen over higher values for efficiency sake. A table with the results of this comparative analysis of the k is presented in Appendix B.

Accuracy is calculated as shown in Equation 3.1.

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{3.1}$$

where tp stands for true positives, tn for true negatives, fp for false positives and fn for false negatives. The definitions of these terms are best understood with the help of a confusion matrix

		Ground Truth	
		c_1	$\bar{c_1}$
Classifier	c_1	True Positives	False Positives
	$\bar{c_1}$	False Negatives	True Negatives

Table 3.1: Confusion Matrix.

shown in Table 3.1. Across the top of this table are the observed class labels (ground truth), while along the left side are the classes predicted by the classifier and each cell contains a count of the number of predictions made by the classifier that match the appropriate class label.

Statistical significance is reported at 95% using paired t-Test. Standard pre-processing operations i.e. lemmatisation and stopwords removal are also applied to all datasets. Feature selection using χ^2 metric is used to limit our term-document space to the top 100 most informative terms for the incident reports datasets (due to their smaller vocabulary size) and top 300 terms for all other datasets.

3.1.2 Semantic Indexing using Knowledge-resource-based Approaches

In this section we present classification results of semantic indexing using WordNet for providing semantic relatedness. We include in our comparison, semantic relatedness computed using Wu & Palmer, Lin, Leacock & Chodorow and Jiang & Conrath metrics (see Section 2.1.1). Accordingly, we compare the following representations:

- BASE- Baseline VSM representation, no semantic indexing
- WUP - Semantic indexing with semantic relatedness computed using Wu & Palmer metric
- LIN - Semantic indexing with semantic relatedness computed using Lin metric
- LCH - Semantic indexing with semantic relatedness computed using Leacock & Chodorow metric
- JCN - Semantic indexing with semantic relatedness computed using Jiang & Conrath metric

Text classification results are presented in Table 3.2. Results presented with a '+' sign represent a statistically significant improvement compared to the baseline (BASE) and best result in each row is presented in bold. Values with the '-' represent a statistically significant decline in

Dataset	BASE	WUP	LIN	LCH	JCN
Ohsumed					
BactV	85.1	76.3-	77.2-	**86.2+**	85.2
CardR	90.0	82.8-	84.2-	**90.6**	89.2
NervI	**91.4**	83.4-	85.1-	90.4-	88.9-
MouthJ	**89.9**	86.4-	89.8	83.1	88.7-
NeopE	91.6	85.9-	91.2	84.1	**91.9**
DigNut	87.8	80.3-	**88.4**	77.7	**88.4**
MuscS	83.1	76.7-	83.2	73.5	**84.0**
EndoH	**91.4**	79.4-	89.1-	75.6	91.3
MaleF	**92.3**	86.9-	**92.3**	85.2	92.0
PregN	**89.7**	80.3-	87.4-	78.2	87.5-
ImmunoV	**78.7**	72.6-	77.7	70.1	76.5-
NervM	84.5	81.4-	83.9	76.0	**85.5**
RespENT	87.2	77.4-	**87.7**	72.6	**87.7**
20 Newsgroups					
Hardw	89.8	87.9-	88.2-	**90.4**	90.1
MedSp	**95.9**	89.6-	89.4-	95.2	95.3
CryptE	**95.9**	71.6-	71.5-	92.7-	95.4
ChrisM	88.9	81.9-	88.5	79.4	**89.0**
MeastM	95.1	89.3-	94.8	82.8	**95.2**
GunsM	**93.4**	85.5-	91.4-	82.5	92.3-
AutoC	94.4	83.8-	93.3	80.1	**94.7**
Reuters					
StratM	**88.8**	77.5-	76.2-	85.5-	88.0
EntTour	**94.7**	82.5-	85-	92.7-	94.6
EqtyB	**95.7**	83.3-	86.3-	93.7	**95.7**
FundA	90.3	77.0-	76.5-	85.2-	**90.8**
InRelD	**92.6**	82.1-	83.6-	90.7-	92.2
NewProdRes	**85.9**	74.5-	75.5-	81.7-	85
ProdNP	**87.4**	79.2-	79.8-	83.7-	86.7
OilGas	**87.8**	80.5-	80.4-	85.4	85.7
ElectG	**88.7**	76.1-	78-	86.2-	87.8
Incident Reports					
Fire	84.4	**88.5+**	87.5+	83.5-	85.0
Collision	82.2	75.5-	77.5-	81-	**83.0**
Rollover	79.8	73.5	78.5	80.0	78.0
CollRoll	86.5	81.5-	83.5-	**87.0**	85.5
MiscInc	**84.0**	83.5	81.5-	82.0-	78.0-
CraneFP	**87.5**	72-	71-	**87.5**	84.5
ShovFP	**88.3**	75.0-	77.0-	84.5	85.0
Movie Reviews					
MovieRev	71.3	68.3-	67.7-	69.1-	**71.4**

Table 3.2: Classification accuracy of semantic indexing using knowledge-based approaches.

text classification performance compared to BASE. As can be observed, very few improvements are realised from the knowledge-resource based approaches. In fact, most results are statistically significantly worse than BASE. Jiang & Conrath (JCN) has been shown to provide better perfor-

mance than the other WordNet metrics in NLP tasks such as synonymy detection (Budanitsky & Hirst 2006). This is reflected in our results where JCN performs best out of all WordNet based semantic relatedness approaches. Nonetheless, JCN produces no significant improvement over BASE.

The poor performance of WordNet based approaches on text classification is likely due to the fact that, being external to the training corpus, WordNet does not reflect the relatedness between terms that is suited to the discriminatory semantics of the target corpus. Take for example the BactViral dataset. This dataset contains documents related to bacterial diseases in one class and those related to viral diseases in the other. Note that many of the terms in both classes will be about diseases, medications, symptoms and other medical vocabulary. Therefore, WordNet is likely to establish strong semantic relatedness among terms from different classes because WordNet is actually ignorant of the class divide existing in the corpus. For example, the LIN metric assigns a similarity value of 0.82 out of a maximum of 1.0 to the terms 'bacteria' and 'virus'. Given that these two terms belong to different classes and are important for discriminating between the two classes, it is easy to see how assigning a high similarity value to the term pair can quickly blur the class distinction between documents.

In addition, word sense ambiguity and vocabulary coverage are likely to have an adverse effect on the performance of semantic relatedness computation using WordNet. For example, across all the datasets, an average of over 20% of the terms from the indexing vocabulary are missing from WordNet. Also, the average number of senses per term across all datasets is 4.6 with some datasets having an average of over 6 senses per term. All these are likely to lead to noisy semantic relatedness values between terms. In our approach for computing semantic relatedness using WordNet, we do not employ word-sense disambiguation. Rather, we adopt the popular approach of taking the maximum relatedness between any combination of the senses of the two given terms as described in (Budanitsky & Hirst 2006).

Because of the poor results achieved using WordNet based semantic relatedness approaches, we do not take this class of approaches any further in this thesis.

3.1.3 Semantic Indexing using Distributional Approaches

In this section, we present a comparative analysis of semantic indexing with semantic relatedness computed using distributional approaches. We include three distributional semantic relatedness

approaches in our study: document co-occurrence, NPMI and LSI (see Section 2.1.2). All experiments with LSI in this thesis are performed using the JAMA matrix package [1]. Here also, the GVSM is used for semantic indexing by providing semantic relatedness computed using the respective approach. Accordingly, we compare the following representations:

- BASE: Baseline BOW approach without term relatedness
- DOCCOOC: Term relatedness estimated from document co-occurrence
- NPMI: Term relatedness calculated using Normalised Positive Pointwise Mutual Information
- LSI: Term relatedness estimated from latent semantic analysis

Classification accuracies are presented in Table 3.3. Again, values with the '+' sign represent a significant improvement in text classification accuracy compared to the baseline and '-' represent a significant decline in classification accuracy and best results in each row presented in bold. In comparison with the knowledge-resource based approaches (see Table 3.2) much significant improvement in text classfication accuracy has been achieved using distributional semantic relatedness approaches. Semantic indexing using these distributional approaches has resulted in statistically significant improvement in 45.95% of the datasets using DOCCOOC, 43.24% using LSI and 45.95% using NPMI. However, on many other datasets, text classification performance was not improved by semantic indexing. On some datasets, semantic indexing has even led to a decline in classification accuracy. For example, semantic indexing using DOCCOOC resulted in a significant drop in classification accuracy on 4 datasets, *MedSp*, *CryptE*, *OilGas* and *ElectG*, while no significant improvement was realised from DOCCOOC on 16 datasets e.g. *MeastM*, *GunsM*, *AutoC* and *BaseH*. Similarly, NPMI and LSI also performed significantly worse than BASE on 10 and 4 datasets respectively and produced no significant improvement on 10 and 17 datasets respectively.

Overall, the datasets created from the Ohsumed corpus benefited the most from semantic indexing. The Ohsumed corpus, being a collection of academic abstracts, contains the most professionally written documents of all other corpora. This means that noise from misspellings is likely to be minimal on this group of datasets. The style of the documents in this corpus is also consistent

[1] http://math.nist.gov/javanumerics/jama/

Dataset	BASE	DOCCOOC	NPMI	LSI
Ohsumed				
BactV	85.1	88.6^{+}	$\mathbf{90.0}^{+}$	87.5^{+}
CardR	90.0	92.2^{+}	$\mathbf{93.8}^{+}$	90.7
NervI	91.4	91.0	$\mathbf{92.9}^{+}$	90.5
MouthJ	89.9	92.2^{+}	$\mathbf{92.9}^{+}$	92.0^{+}
NeopE	91.6	93.8^{+}	$\mathbf{94.2}^{+}$	94.0^{+}
DigNut	87.8	91.3^{+}	$\mathbf{93.2}^{+}$	91.5^{+}
MuscS	83.1	87.0^{+}	$\mathbf{91.1}^{+}$	86.5^{+}
EndoH	91.4	95.8^{+}	$\mathbf{96.5}^{+}$	95.4^{+}
MaleF	92.3	94.9^{+}	$\mathbf{95.6}^{+}$	95.1^{+}
PregN	89.7	90.4	$\mathbf{90.9}^{+}$	90.4
ImmunoV	78.7	82.5^{+}	$\mathbf{84.8}^{+}$	82.7^{+}
NervM	84.5	88.1^{+}	$\mathbf{91.0}^{+}$	87.8^{+}
RespENT	87.2	88.1	$\mathbf{91.0}^{+}$	88.3
20 Newsgroups				
Hardw	89.8	90.9^{+}	$\mathbf{91.2}^{+}$	90.3^{+}
MedSp	**95.9**	93.8^{-}	95.8	93.6^{-}
CryptE	**95.9**	90.3^{-}	91.8^{-}	90.6^{-}
ChrisM	88.9	$\mathbf{90.5}^{+}$	89.9^{+}	$\mathbf{90.5}^{+}$
MeastM	95.1	**95.3**	94.9	**95.3**
GunsM	93.4	**94.0**	**94.0**	**94.0**
AutoC	94.4	95.1	$\mathbf{96.2}^{+}$	95.0
Reuters				
StratM	88.8	89.4	83.7^{-}	**89.6**
EntTour	94.7	$\mathbf{95.7}^{+}$	95.3	95.6^{+}
EqtyB	95.7	95.5	94.8^{-}	95.6
FundA	90.3	92.0^{+}	89.9	$\mathbf{92.1}^{+}$
InRelD	92.6	94.1^{+}	91.7	$\mathbf{94.3}^{+}$
NProdRes	85.9	86.9	80.4^{-}	**86.7**
ProdNP	87.4	$\mathbf{89.3}^{+}$	88.4	88.9^{+}
OilGas	**87.8**	86.3^{-}	85.7^{-}	86.2^{-}
ElectG	**88.7**	84.6^{-}	84.0^{-}	84.5^{-}
Incident Reports				
Fire	84.4	**87.0**	85.8	86.9
Collision	**82.2**	80.9	76.8^{-}	81.3
Rollover	**79.8**	79.1	77.7	78.2
CollRoll	**86.5**	83.6	80.5^{-}	84.3
MiscInc	84.0	**84.1**	82.0	**84.1**
CraneFP	87.5	**88.3**	82.4^{-}	87.9
ShovFP	**88.3**	86.6	**88.3**	83.8^{-}
Movie Reviews				
MovieRev	71.3	78.6^{+}	81.8^{+}	$\mathbf{79.3}^{+}$

Table 3.3: Classification accuracy of semantic indexing using distributional appraoches.

i.e. unlike the Reuters corpus where sometimes, the entire content of a document is a single table of values which provides little benefit for semantic indexing. Semantic indexing has also proven

beneficial on the Movie Reviews (MovieRev) dataset. This demonstrates the utility of semantic indexing for sentiment classification. Semantic indexing has not produced much improvements on the other dataset groups (20 Newsgroups, Reuters and Incident Reports). This is likely due to the noise in the datasets produced by the informal writing style of documents, and inconsistency in the format of documents. It thus seems evident that clean, formal documents are important for the performance of semantic indexing. This finding is in contrast to (Xue & Zhou 2006) where more informal documents were found to benefit more from their distributional features approach. This is not surprising because distributional features are quite different from distributional semantic relatedness. Distributional features propose replacing frequency counts in document vector representations with measures of first appearance and compactness of terms within a document. In contrast, the aim of distributional semantic relatedness is to model the semantic relationship between pairs of terms based on the co-occurrence of these terms in the corpus.

Table 3.3 also suggests a relationship between the length of documents as well as the size of datasets, and the performance of semantic indexing. Note that no significant improvement from semantic indexing is observed on the incident reports group of datasets which have 200 documents per dataset, compared to the other groups that have 1000 documents per dataset (see Section 2.6 for more details on the datasets). This is perhaps because the sizes of these datasets do not allow for learning beneficial semantic relatedness knowledge from co-occurrece statistics. Contrast this with all other corpora where at least some significant improvement is observed.

Distributional semantic relatedness approaches sometime fail because of their tendency to occasionally establish relationships that are too general and hence not very discriminatory. For example in the BactViral dataset, the terms "biopsy" and "treat" co-occur 10 times which indicates a strong relationship. However, the two words co-occur almost equally across class boundaries which means that the relationship between them is a weak indicator of class membership. In contrast, the words "endoscopy" and "helicobacter" co-occur 5 times, all within the **Bacterial** class which makes this relationship a stronger indicator of class membership. Because of the higher co-occurrence frequency between "biopsy" and "treat", the semantic relation between them is likely to be stronger than the relation between "endoscopy" and "helicobacter".

Considering the additional cost of acquiring term-relatedness, it is important to empirically determine when it is beneficial to use semantic relatedness in text retrieval. In the next section, we explore the use of meta-learning for predicting, given any dataset, whether or not to apply

semantic indexing.

3.2 Predicting When to use Semantic Indexing

To be able to predict when and when not to use semantic indexing, we turn to meta-learning. The primary goal of meta-learning is to produce proper guidance on the right algorithm to use, from a number of available algorithms and techniques, according to the nature of the problem. (Vilalta et al. 2004). Much work has been done in the area of meta-learning. For example a meta-learner to recommend the appropriate classifier given a dataset is presented in (Bensusan, Giraud-Carrier & Kennedy 2000). We wish to use meta-learning to recommend, given a dataset, whether or not to use semantic indexing.

Given that we already have a rich collection of datasets for which we know the performance of semantic indexing, we would ideally like to use a supervised meta-learning approach. In machine learning, framing problems as supervised tasks makes it easier to achieve higher levels of performance. Thus, given a dataset, we would like our meta-learner to assign to that dataset the binary decision of whether or not to use semantic indexing, using a model learned from a collection of datasets for which we have prior knowledge of the performance of semantic indexing. Ideally, semantic indexing should only be used if doing so will lead to significant improvement in text classification performance compared to not using semantic indexing. Thus our training dataset will consist of instances (datasets) labeled with the decision to use semantic indexing if semantic indexing produced a significant improvement in text classification performance, and the decision not to use semantic indexing if semantic produced no significant improvement.

To develop our meta-learning system, we decided to use case-based reasoning . Case-based reasoning is a problem solving methodology where, given a new problem, the solution of the most similar case from a database of previously solved cases is adapted for solving the new problem. Similarity between cases is computed by computing the similarity between the attributes of these cases. Case-based reasoning is suitable for our task based on the assumption that datasets for which semantic indexing does not work are likely to share some attributes in common. Hence, we expect the decision (whether or not to use semantic indexing), that applied to the most similar cases to a given problem, to be suitable for the new problem dataset.

Case-based reasoning has been widely adopted for developing meta-learning systems. For ex-

ample, a meta case-based technique for selecting case-base maintenance algorithms is presented in (Cummins & Bridge 2011). In this approach, an individual meta-case models an entire case-base where the case solution is the maintenance algorithm that provides the best performance on that case-base and the case description comprises a set of attributes that are derived using dataset complexity metrics. Another case-based approach for selecting the best sentiment lexicon given a sentiment classification dataset is presented in (Ohana, Delany & Tierney 2012). Here also, a dataset is represented as a single case where the case solution is the best performing sentiment lexicon for the dataset. The case description is modelled as an n-dimensional feature vector derived from document, sentence and term-level statistics of as well counts of part-of-speech information and punctuations. The attributes chosen for case representation are designed to capture the subjectivity of the corresponding dataset. Another system is presented in (Lindner & Studer 1999) which uses a case-based approach to select the best classification algorithm for a dataset. The datasets considered in this work are not limited to textual datasets, thus, the attributes used for case representation are designed to capture characteristics of datasets that contain both numeric and symbolic attributes.

3.2.1 Case-Based Prediction Framework

Figure 3.1 shows both the training and test phases of our case-based system. Given a collection of training datasets, the case generator creates a case representation for each dataset. The case description comprises a set of nine attributes a_1 to a_n (discussed in Section 3.2.2) that capture the properties of the dataset. The case solution is a binary judgement of whether or not to apply semantic indexing to the dataset. A case is labelled with the solution to use semantic indexing (Sem) if the improvement from applying semantic indexing is statistically significant. Otherwise, we label the case with the decision not to use semantic indexing ($\neg$Sem). For example for the DOCCOOC technique, semantic indexing produced a significant improvement on the Hardware with accuracy of 90.9% compared to BASE (89.9%) (see table 3.3) and the decision to use semantic indexing is selected as the case solution for Hardware. On the other hand on the *MedSpace* dataset, DOCCOOC produced a decline of 2.1 % in accuracy and thus the solution for this case is not to use semantic indexing. For computing similarity between cases, we use the Manhattan distance,

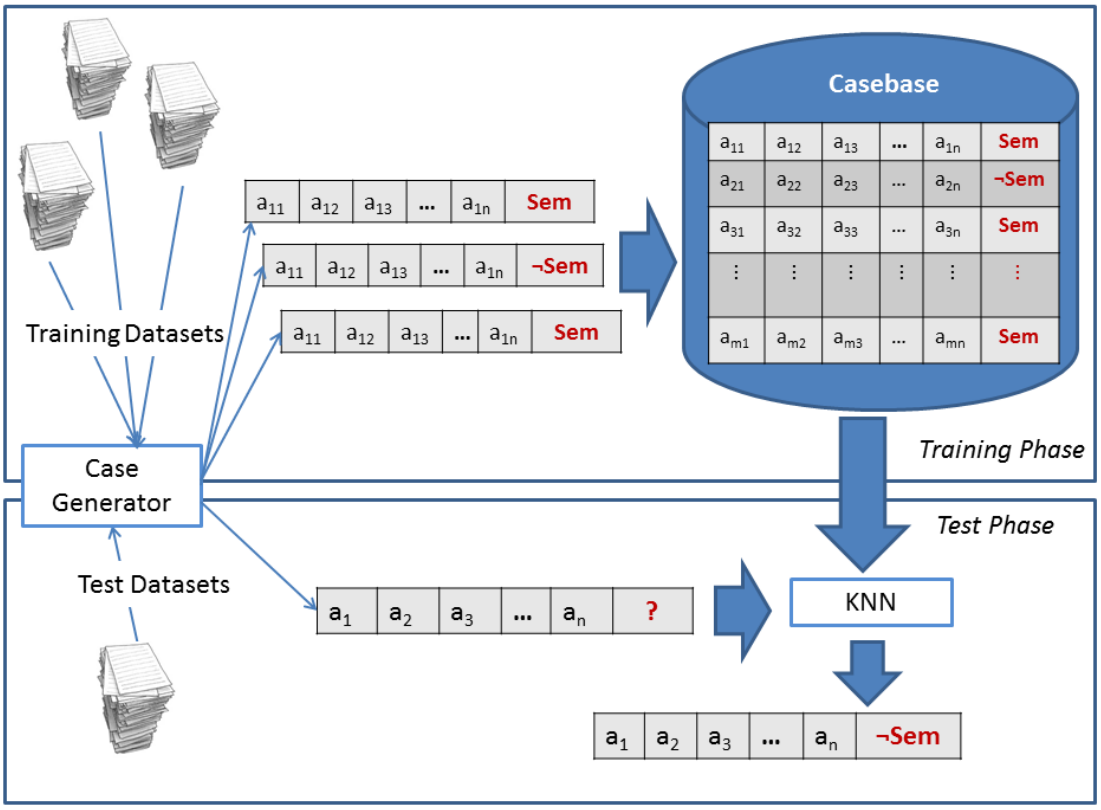

Figure 3.1: Case-based approach using dataset meta-data to predict when to use semantic indexing.

given in equation 3.2, as a simple, baseline similarity function.

$$Dist(a,b) = \sum_{i=1}^{N}(|a_i - b_i|) \tag{3.2}$$

In the next section, we discuss the set of attributes used for case representation.

3.2.2 Dataset Attributes

Several different attributes have been considered in previous works for capturing the characteristics of datasets. A common baseline approach is presented in (Lindner & Studer 1999) where several statistical measures are used to characterise datasets. Note that no motivation is given for the choice of characteristics or meta-attributes. The meta-attributes used include number of instances, number of features of the dataset, ratio of symbolic features, number of classes, default error rate, standard deviation of class distribution, relative probability of defective instances, number of records with missing values, relative probability of missing values and number of missing values.

Attribute Name	Description
AveTermCount	Average number of terms per document
MaxDocFreq	Maximum term document frequency
AveDocFreq	Average term document frequency
MaxIDF	Maximum term Inverse Document Frequency
AveIDF	Average term Inverse Document Frequency
Nearest Neighbour Similarity	Average similarity of nearest neighbours
AveNSim	Average neighbourhood similarity
MinNSim	Minimum neighbourhood similarity
MaxNSim	Maximum neighbourhood similarity

Table 3.4: Summary of dataset attributes used for meta-case representation.

Note that all of these meta-attributes are not useful for our task of predicting the performance of semantic indexing on text datasets. For example, term-document matrices are typically sparse with most feature values missing in any one document. Thus, it is unlikely that the measure of missing values is a good indicator of the performance of semantic indexing. Also, the number of instances and attributes are the same for all datasets, except the incident report datasets. Thus, these are also excluded from consideration as features. The authors also propose additional information theoretic features which are only applicable to symbolic dataset features and thus are not applicable for text datasets.

The authors in (Peng, Flach, Soares & Brazdil 2002), propose using meta-attributes created from measuring the characteristics of decision trees generated from the datasets. Here also, no justification was given for this choice of meta-attributes. This approach involves generating a decision tree from the dataset and then measuring attributes such as the number of nodes, number of branches and height of the decision tree. Given that our classifier of choice is kNN, it is not clear how useful the characteristics of a decision tree will be at predicting the performance of semantic indexing used with kNN.

The work in (Cummins & Bridge 2011) presents a meta learning approach for the selection of case-base maintenance algorithms. The meta-attributes used to characterise case-bases were chosen to model the complexity of these case-bases as case-base complexity is seen as the important predictor of the performance of case-base maintenance algorithms. The meta-attributes considered are divided into three categories: Measures of Overlap of Attribute Values, Measures of Separability of Classes and Measures of Geometry, Topology and Density of Manifolds. Note that all the meta-attributes in the three categories are supervised, meaning that the class labels

of data instances (documents in our case) need to be considered. However, recall that the VSM and semantic indexing are not limited to supervised tasks. On the contrary, both the VSM and semantic indexing were originally designed for unsupervised document retrieval. Accordingly, it is highly desirable to consider unsupervised meta-attributes that are applicable for both supervised and unsupervised tasks.

Considering the limitations of the meta-attributes proposed in previous works, and the lack of strong motivation behind them, we propose a new set of meta-attributes. Recall that semantic indexing is applied to the term-document space representation of a document collection and not the actual document collection itself. Thus when selecting meta-attributes, we choose the types of attributes that are typically used for creating vector representations of documents e.g. term frequency and inverse document frequency. Also, because our classifier of choice is kNN, we use attributes that describe the neighbourhood structure of the datasets. A summary of the attributes we consider is presented in table 3.4. We describe these attributes in detail in the following sections. A table of the attributes and corresponding values used in our experiments is provided in Appendix D.

Average Terms Per Document

This is a measure of the average number of terms per document which is calculated after text preprocessing: stopwords removal, term normalisation and feature selection. Thus, the count of terms in a document is restricted to the terms from the indexing vocabulary. This is calculated as shown in equation 3.3.

$$TermCount(d_i) = \sum_{t_j \in T} d_i \tag{3.3}$$

Where t_i is a term in document d_i and T is the entire indexing vocabulary. The average term count for the entire dataset is calculated by taking the average term count for all documents in the dataset as in equation 3.4.

$$AveTermCount = \frac{\sum_{d_i \in D} TermCount(d_i)}{|D|} \tag{3.4}$$

Document Frequency

The document frequency of a term t_i is a count of the number of documents in which t_i occurs. Document frequency is often used as a feature selection technique under the premise that very rare terms are not informative and thus do not contribute much to document retrieval. At the same time, terms that appear in almost all documents are also not very discriminatory and can be considered noisy in the term document space. Such high frequency terms are also likely to co-occur with almost every other term thus polluting the generalisation process. Hence we utilise two metrics to measure the effect of document frequency: **Maximum DF (MaxDocFreq)** which is the maximum document frequency over all terms and **Ave. DF (AveDocFreq)** which is the average document frequency of over all terms.

Inverse Document Frequency

Inverse Document Frequency (IDF) is a function designed to give a weighting inversely proportional to the document frequency of terms. IDF captures the premise that terms with very high document frequency are less informative than terms that occur less often. The formula for IDF is given in equation 3.5 where N is the total number of documents and $df(t)$ is the document frequency of t.

$$IDF(t) = log_2 \frac{N}{df(t)} \tag{3.5}$$

We use the **Maximum IDF (MaxIDF)** and the **Average IDF (AveIDF)** to obtain a measure of rare terms in our datasets.

Nearest Neighbour Similarity

We measure the tightness of the clustering of documents in a dataset using the distance between each document, and the other documents in its neighbourhood as shown in Figure 3.2. Nearest Neighbour Similarity of a document d_j is calculated by iteratively retrieving successively larger neighbourhoods k of d_j up to the neighbourhood size K (we use $K = 10$) and computing the similarity between d_j and all documents in its neighbourhood. This is shown in equation 3.6.

$$P_k(d_j) = \frac{\sum_{i=1}^{k} Sim(d_j, d_i)}{k} \tag{3.6}$$

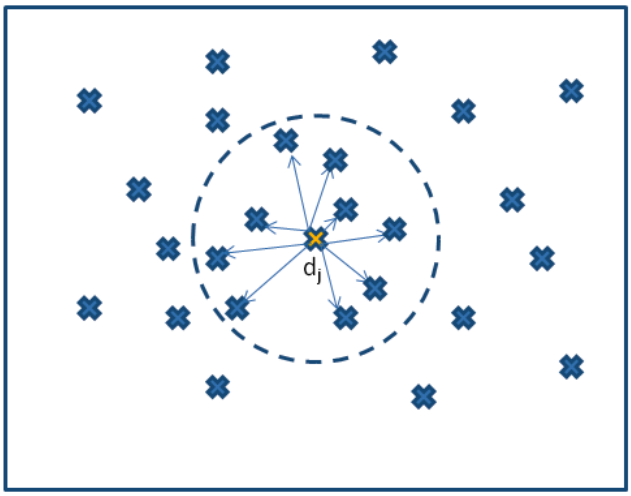

Figure 3.2: Nearest Neighbour Similarity calculated using the distance of a target document d_j to its k nearest neighbours.

Where $Sim(d_j, d_i)$ is the cosine similarity between document d_j and d_i. The final Nearest Neighbour Similarity measure for the entire dataset is computed as the average Nearest Neighbour Similarity of all documents d_j.

Neighbourhood Similarity

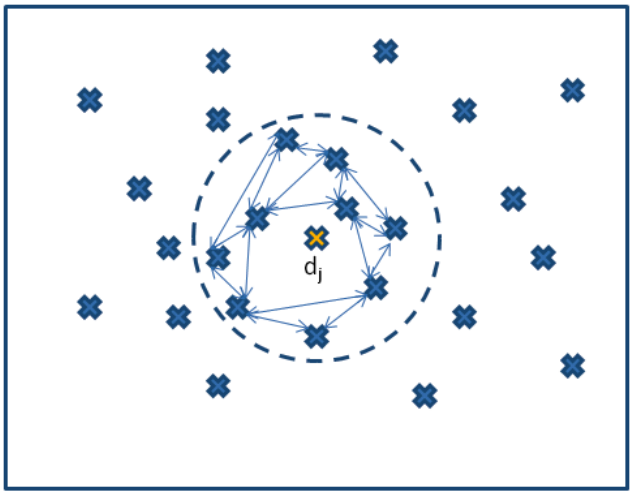

Figure 3.3: Neighbourhood similarity of document d_j measures using the distance between k nearest neighbours of d_j.

While Nearest Neighbour Similarity measures the distance between a target document and its

nearest neighbours, this metric calculates the average pair-wise similarity between all k nearest neighbours of the target document d_j as shown in Figure 3.3. We use a neighbourhood size of $k = 10$. We then calculate the average, minimum and maximum neighbourhood similarity over all documents to obtain the **Average Neighbourhood Similarity (AveNSim)**, **Minimum Neighbourhood Similarity (MinNSim)** and **Maximum Neighbourhood Similarity (MaxNSim)** respectively for that dataset.

The average similarity between the nearest neighbours of a document tells us how tightly clustered the neighbourhood of that document is. In turn, the aggregation over all documents provides us with information about how tightly clustered documents are in the entire term document space.

3.2.3 Evaluation

The aim of this evaluation is to determine how well our meta case-based approach (CBR) predicts when and when not to use semantic indexing for text representation. We compare this with a baseline approach (BASELINE) that always applies semantic indexing. Our hypothesis is that our case-based approach should be able to identify datasets that are not likely to benefit from semantic indexing which allows for applying semantic indexing to datasets in a systematic fashion. Accordingly, we treat this as a classification task where accuracy is measured as the percentage of test cases that are labelled with the correct decision (to generalise or not). We report the classification accuracy over a leave-one-out validation using a 3-NN approach.

	Overall	DOCCOOC	NPMI	LSI
BASELINE	55.81	41.86	46.51	37.21
CBR	**79.07**	**81.4**	**88.37**	**72.09**
CBR+	**86.05**	**86.05**	**93.02**	**79.07**

Table 3.5: Classification accuracy of predicting when to use semantic representation.

From the results shown in Table 3.5, it is clear that our meta case-based system predicts when to apply semantic indexing with high accuracy. The results in the **Overall** column represent the accuracy of our prediction across all semantic indexing techniques. That is, deciding to use semantic indexing always, using the best semantic indexing approach, we match all datasets that are labelled with the decision to use semantic indexing (55.81%) but we also apply semantic indexing to many other datasets (44.19%) that that do not benefit from semantic indexing. However, using our case-based approach, we selectively apply semantic indexing to datasets only when we should,

Parameter	Value
Encoding	Integer
Genotype Range of Values	0 - 10
Individual Length	9
Population Size	100
Selection Strategy	Tournament

Table 3.6: Genetic Algorithm Parameter Settings.

and avoid doing so when we should not with accuracy of 79.07% . The other columns (DOCCOOC, NPMI and LSI) provide a break-down of our performance for each individual semantic relatedness technique respectively.

The CBR+ row shows results of the Case-Based approach with optimal weights learned for the meta-case attributes using a Genetic Algorithm (GA) where the set of weights used range from 0 to 10. A comprehensive review of applying weighting to kNN retrieval is provided in (Wettschereck, Aha & Mohri 1997). GA's are computational search heuristics that mimic the process of natural selection. In a GA, a population of candidate solutions called individuals are evolved towards an ideal solution over generations, using mechanisms such as selection, inheritance, mutation and crossover. For our GA implementation, we use an integer encoding with values in the range 0 to 10, and an individual length of nine to represent the weights and attributes respectively. Each attribute of an individual is referred to as a Genotype. Additional parameter settings include a population size of a hundred and a tournament selection strategy. These parameter settings are provided in Table 3.6.

From these results we can see that our set of attributes are predictive of the effectiveness of applying semantic relatedness for text representation.

The weights learned for our attributes by the genetic algorithm can be divided into high, **Nearest Neighbour Similarity**; medium, **MaxIDF**, **Ave. Tokens Per Doc.**, **MaxDocFreq** and **MaxNSim**; and low, **AveDocFreq**, **AveIDF**, **AveNSim** and **MinNSim**. The high weight assigned to Nearest Neighbour Similarity indicates the importance of the similarity between documents in a dataset in determining the performance of semantic indexing. Note that lower values of Nearest Neighbour Similarity indicate higher variation in vocabulary in these datasets. This indicates that better semantic relatedness can be extracted from datasets that have less variable vocabulary indicated by a higher Nearest Neighbour Similarity. Higher variation in indexing vocabulary of these datasets can be attributed to their short length. However, in general, other factors such as informal

language and inconsistency in document style can contribute to increasing variation in indexing vocabulary. Note that although this attribute is important for determining the performance of semantic indexing, the performance of semantic indexing is not dependent exclusively on this single attribute. From Table D.1 in Appendix D, we can see that there are datasets with similar Nearest Neighbour Similarity that have contrasting performance with respect to semantic indexing. Hence, the use of meta-learning allows us to leverage the other attributes to improve the accuracy of our prediction.

3.3 Chapter Summary

In this chapter, we investigated the benefit of semantic indexing for text classification. We used the GVSM framework in our study to test four different knowledge-resource based approaches and three different distributional approaches for computing semantic relatedness. The performance of the semantic indexing with the knowledge-resource-based approaches showed very little improvement with many of the results being significantly worse than not using semantic indexing. Note that while these WordNet based metrics have been widely evaluated on linguistic tasks such as synonymy detection and word pair association, to the best of our knowledge, this is the first time such a comprehensive evaluation has been reported using these metrics on text classification.

In contrast however, distributional approaches showed more potential for semantic indexing with substantial gains in text classification performance. However, the performance of distributional semantic relatedness approaches also revealed that semantic indexing does not always improve text classification performance and may sometimes even be harmful. Our results suggest that datasets with documents written in a more professional and consistent style benefit more from semantic indexing. We also observed that datasets with fewer and shorter documents benefited less from semantic indexing.

Considering that semantic indexing introduces additional overhead to the process of text representation, we set out to determine when and when not to apply semantic indexing using meta-learning. Accordingly, we presented a case-based approach for predicting when to use semantic indexing. Results show that our case-based approach is able to correctly predict the performance of semantic indexing on a range of datasets with over 80% accuracy. Again, to the best of our knowledge, this is the first time any attempt has been made to predict when to apply semantic

indexing.

An important consideration when building a case-based system is the choice of attributes for case representation. The attributes we used were obtained from several statistical metrics that capture various important characteristics of text datasets. These range from statistics of document frequencies of terms to measures of clustering of document neighbourhood. The high accuracy achieved in predicting when to use semantic indexing indicates that the attributes used for case representation capture characteristics of text datasets that are predictive of the performance of semantic indexing.

We further used a genetic algorithm to learn the relative importance of our attributes. The high weight assigned to the Nearest Neighbour Similarity attribute indicates the importance of the structure of a dataset is in determining the performance of semantic indexing. From Table D.1 in Appendix D, we observe that the incident report datasets for which semantic indexing did not work, all datasets had a much lower **Nearest Neighbour Similarity** compared to the other datasets. This implies that for the incident report datasets in particular, the sparseness in the datasets affected the quality of semantic relatedness extracted. Sparseness in these datasets can be attributed to the short length of the documents which means that any one document contains only a few terms from the vocabulary, thereby reducing the similarity between documents.

Chapter 4

Relevance Weighted Semantic Indexing

Semantic indexing has not resulted in consistent improvement in text classification performance. Our intuition on this is that the semantic indexing process does not properly capture the relevance of terms in document representations. It is well known that all terms in a corpus do not have the same importance with some terms being better at discriminating between classes, making them more relevant to the classification task. For example, to identify documents that belong to the class *Sports* , the terms "goal", "match", "team" and "football" are more relevant than terms like "rain", "happy" and "glass". Thus, it is important for semantic indexing that such class-indicative terms are recognised and assigned higher importance or weight in document representations. While semantic indexing captures the semantic relatedness between terms, we argue that it is not good at capturing the class-indicativeness or relevance of terms.

In this chapter, we introduce a novel framework called Relevance Weighted Semantic Indexing (RWSI) which extends the GVSM by capturing both local (within-document) and global (collection-wide) term relevance for semantic indexing. Global relevance of terms can be learned directly from the training corpus using supervised term weighting functions.

A second aim of this chapter is to demonstrate the utility of supervised indexing for text classification. Accordingly, we demonstrate how the RWSI framework can be used exclusively for supervised document indexing, using an approach we call Relevance Weighted Indexing (RWI). A comparative evaluation of our RWI with the standard tf-$\delta(t)$(see Section 2.4) approach shows RWI to lead much more consistent improvement in text classification performance.

This chapter is organised as follows: in Section 4.1 we provide a detailed analysis of the

inner workings of the GVSM. In Section 4.2 we present an analysis of how term weights can be adversely affected by semantic indexing and demonstrate how this can be addressed using vector normalisation. In Section 4.3 we highlight the need for relevance weighting and present the RWSI framework which extends the GVSM framework by introducing relevance weights of terms for semantic indexing. In Section 4.5, we demonstrate the RWI approach which utilises the RWSI framework for supervised document indexing. Evaluations are presented in Section 4.6. We conclude this chapter with a summary in Section 4.7.

4.1 Analysis of GVSM

The traditional vector space model (VSM) assumes independence between terms. However, this independence assumption is an over simplification because different terms within an indexing vocabulary often have related or even identical meanings. The implication of the term independence assumption is that the similarity between related documents can only be correctly estimated if these documents share the exact same lexical terms. The GVSM framework was proposed for capturing the relevant dependencies between term in document representations (Wong et al. 1987). In this section, we provide a comprehensive analysis of semantic indexing using the GVSM. In Section 2.2.4 we formally presented the GVSM. For the sake of completeness, we repeat some of the mathematical equations that are the basis for the GVSM. Given any two documents q and d, their similarity can be computed in the GVSM as:

$$Sim(q,d) = \sum_{i}^{n}\sum_{j}^{n} u_i \vec{t_i} w_j \vec{t_j} \tag{4.1}$$

Where n is the dimension of the vector space (i.e. the number of terms in the indexing vocabulary), u_i and w_j are the initial (*tf-idf*, binary e.t.c.) weights for the terms t_i and t_j in the query q and document d respectively, and $\vec{t_i}$ and $\vec{t_j}$ are vector representations of t_i and t_j respectively. The product of the two term vectors, $\vec{t_i}$ and $\vec{t_j}$, provides the relatedness between the corresponding terms t_i and t_j. Thus, the product of the two term vectors, $\vec{t_i}$ and $\vec{t_j}$, in Equation 4.1 can be replaced with the a function, $Rel(t_i, t_j)$, that returns the relatedness between terms t_i and t_j.

Accordingly equation 4.1 can be rewritten as follows:

$$Sim(q,d) = \sum_{i}^{n}\sum_{j}^{n} u_i w_j Rel(t_i,t_j) \tag{4.2}$$

$$Sim(q,d) = \sum_{i}^{n} u_i \sum_{j}^{n} w_j Rel(t_i,t_j) \tag{4.3}$$

Introducing the function $Rel(t_i,t_j)$ allows for using any approach for computing the relatedness between terms t_i and t_j without restricting to the vector product of term vectors. Recall that document d is represented as a vector $\vec{d}$ in euclidean space with dimension the size of the vocabulary V as shown in Equation 4.4.

$$\vec{d} = (w_1, w_2, ..., w_n) \tag{4.4}$$

Where the corresponding weight, $w_i \in \vec{d}$, of each term $t_i \in V$ is non-zero only if t_i occurs in d, and zero otherwise. The same applies for $\vec{q}$. Therefore, from Equation 4.3, for each term $t_i \in V$, the original weight of t_i in $\vec{d}$ (including zero weight if t_i is absent in d) is replaced by $\sum_j^n w_j Rel(t_i,t_j)$. Accordingly, even if t_i does not occur in d, it now gets a corresponding weight $w_i' = \sum_j^n w_j Rel(t_i,t_j)$ in the new semantic representation of d, if t_i is related to one or more terms $t_j \in d$ with non-zero weight. This is illustrated in Equation 4.5.

$$d' = (\sum_{j}^{n} w_j Rel(t_1,t_j), \sum_{j}^{n} w_j Rel(t_2,t_j), ..., \sum_{j}^{n} w_j Rel(t_n,t_j)) \tag{4.5}$$

$$w_i' = \sum_{j}^{n} w_j Rel(t_i,t_j) \tag{4.6}$$

$$d' = (w_1', w_2', ..., w_n') \tag{4.7}$$

Where w_i' is the new semantic weight of term t_i in d'. Observe from Equation 4.5 that d' is simply the product of the document vector d and an $n \times n$ matrix which we will call T where each entry $\tau_{i,j}$ in T corresponds to the value $Rel(t_i,t_j)$. In other words, the matrix T captures the semantic relatedness of all pairs of terms t_i and t_j in V. Each column j of T correspond to a vector v_j which captures the semantic relatedness of the term t_j and all other terms $t_i \in V$. Document

vectors of the entire collection can also be represented in the form of a document-term matrix which we will call D. Hence from Equation 4.5, the transformation of the entire document-term matrix can be expressed as:

$$D' = D \times T \tag{4.8}$$

Equation 4.8 requires semantic relatedness values to be computed for all pairs of terms t_i and t_j in V, and used to populate the term-term semantic relatedness matrix T. Each vector $\vec{\tau_i} \in T$ provides the semantic relatedness of the corresponding term t_i with all terms $t_j \in V$. Because any term can be at most similar to itself, all entries on the leading diagonal of T (i.e. $i = j$) are consequently assigned a value of 1. Thus, all other entries in T are required to be normalised between 0 and 1, with the value 1 in any cell corresponding to identical term pairs and 0 to dissimilar. The normalisation of the values of T ensures that a term can never be more related to another term than it is to itself. The impact of equation 4.8 will be to boost the presence of related terms that were not contained in the original documents, which in turn has the beneficial effect of making the vector representations of documents that belong to the same class more similar.

4.2 Preserving Local (Within-Document) Relevance

The initial weight w_i assigned to a term t_i in a document d, is designed to reflect the importance or relevance of t_i to d. However, note from Equation 4.5 that the weight w'_i of t_i in the semantic document representation d' is not exclusively determined by the original weight w_i of term t_i. Rather, w'_i is strongly influenced by the weight w_j of the term $t_j \in d$ that t_i is semantically related to, and also by the strength of this semantic relatedness ($Rel(t_i, t_j)$). This means that if t_i is strongly related to many other terms $t_j \in d$, then t_i receives a relatively high weight w'_i, regardless of its original relevance to d. The reverse is also the case, i.e., if t_i is related to only a few terms $t_j \in d$, then t_i receives a relatively low weight. This is certainly an undesired consequence of semantic indexing because, if t_i was initially assigned a relatively low weight w_i due to it being less important or relevant to document d, the aggregation of the semantic relatedness of t_i, if t_i is related to enough other terms, could result in a high weight w'_i in d'. In other words, the relevance of t_i to d is easily lost during semantic indexing, in favour the semantic relatedness between t_i and the terms t_j in d.

This problem of loss of local relevance in term weights is of particular concern in situations where semantic relatedness is computed from corpus co-occurrence statistics. In a typical corpus, any term t_i is likely to have non-zero co-occurrence with many other terms t_j in the collection. Thus, a term t_i which is initially absent, or assigned a low weight in the vector of a document d_j can easily end up having the highest weight after semantic indexing if it co-occurs often with many other terms in the corpus. Hence, by computing term weights as an aggregation of semantic relatedness, the cumulative effect of less important terms can result in significant amounts of noise being added to document representations.

$$D = \begin{array}{c|ccccc} & t_1 & t_2 & t_3 & t_4 & t_5 \\ d_1 & 0.0 & 0.7 & 0.6 & 0.0 & 0.0 \\ d_2 & 0.8 & 0.0 & 0.0 & 0.5 & 0.0 \\ d_3 & 0.1 & 0.9 & 1.0 & 0.0 & 0.0 \\ d_4 & 0.3 & 1.0 & 0.7 & 0.0 & 0.0 \end{array}$$

$$T = \begin{array}{c|ccccc} & t_1 & t_2 & t_3 & t_4 & t_5 \\ t_1 & 1.0 & 0.5 & 0.8 & 0.7 & 0.3 \\ t_2 & 0.5 & 1.0 & 0.2 & 0.2 & 0.3 \\ t_3 & 0.8 & 0.2 & 1.0 & 0.0 & 0.1 \\ t_4 & 0.7 & 0.2 & 0.0 & 1.0 & 0.3 \\ t_5 & 0.3 & 0.3 & 0.1 & 0.3 & 1.0 \end{array}$$

$$D' = \begin{array}{c|ccccc} & t_1 & t_2 & t_3 & t_4 & t_5 \\ d_1 & 0.83 & 0.82 & 0.74 & 0.14 & 0.27 \\ d_2 & 1.15 & 0.5 & 0.64 & 1.06 & 0.39 \\ d_3 & 1.35 & 1.15 & 1.26 & 0.25 & 0.40 \\ d_4 & 1.36 & 1.29 & 1.14 & 0.41 & 0.46 \end{array}$$

Figure 4.1: Example of semantic indexing using the GVSM

We illustrate this point further with the aid of an example. Figure 4.1 shows a sample document-term matrix D with 4 documents and 5 terms, a matrix T which captures the semantic relatedness between all pairs of terms in the vocabulary, and a semantic document-term matrix D' containing semantic document representations derived from D and T using Equation 4.8. Note from Figure 4.1 that document d_1 in D does not contain the term t_1. However after semantic indexing, term t_1 has the highest weight in d'_1. A similar result is seen in d_3 and d_4 where t_1 has low weights of 0.1 and 0.3 respectively. However, after semantic indexing, t_1 again has the highest weight in d'_3 and d'_4 i.e. 1.35 and 1.36 respectively. This happens simply because t_1 is semantically related

to all the terms in d_1, d_3, and d_4. However, t_1 could have been absent or assigned low weights in d_1, d_2 and d_4 because it is not directly important to these documents. For example if these documents had been about cars and t_1 was the term 'Honda', even though 'Honda' is relevant to the topic of cars, it is certainly overrated to think that 'Honda' should be the most important term in d_1, d_3 and d_4, simply because 'Honda' is semantically related to the other terms in these documents. Indeed many documents about cars will have nothing to do with 'Honda'. Likewise, many documents containing the term 'Honda' could also be about the company or motorcycles and have nothing to do with cars. It is clear then that local (within-document) term importance is ignored using the approach in Equation 4.8, resulting in noisy representations. This problem is even more acute in real-world situations where, because of the high dimensionality of document vectors, larger discrepancies can easily result from aggregating semantic relatedness over all terms in a document.

To address the problem of loss of local relevance from semantic relatedness aggregation, we introduce a modification to the approach in Equation 4.8 which is to normalise all row vectors $\vec{d} \in D$ and all column vectors $\vec{\tau} \in T$ to unit length before taking their product. Normalisation is achieved by taking the L2 norm of the corresponding vectors $\vec{d}$ and $\vec{\tau}$. This ensures that the length of the vectors are taken into account i.e. terms that are semantically related to many document terms now get penalised to prevent such terms from dominating document representation. The computation of the L2 norm of a vector v is given in equation 4.9.

$$\| v \| = \sqrt{\Sigma_{i=1}^{n} v_i^2} \tag{4.9}$$

Thus, we can modify Equation 4.8 to reflect this normalisation as follows:

$$D' = D^{rn} \times T^{cn} \tag{4.10}$$

Where D^{rn} is the term document matrix D with all rows L2 normalised, and T^{cn} is the semantic relatedness matrix T with all columns L2 normalised. Figure 4.2 shows the semantic document-term matrix D' from Figure 4.1 with L2 normalisation applied before taking the product of the matrices D and T. Note that the distribution of terms in the document vectors of D' better reflect their original distribution in D. For example t_1 no longer has the highest weight in documents d_1, d_3 and d_4. This highlights the importance of the normalisation function as an es-

$$D' = \begin{array}{c|ccccc} & t_1 & t_2 & t_3 & t_4 & t_5 \\ d_1 & 0.57 & 0.75 & 0.62 & 0.12 & 0.26 \\ d_2 & 0.78 & 0.45 & 0.52 & 0.88 & 0.37 \\ d_3 & 0.64 & 0.72 & 0.72 & 0.14 & 0.26 \\ d_4 & 0.69 & 0.86 & 0.70 & 0.26 & 0.32 \end{array}$$

Figure 4.2: Resulting term-document from Figure 4.1, after semantic indexing with L2 normalisation.

sential component of the semantic indexing process for preserving local (within-document) term importance. Thus, in the remainder of this thesis, the row vectors and column vectors of the D and T matrices respectively are always L2 normalised before matrix multiplication, even if, for the sake of convenience, the superscript notation (D^{rn} and T^{cn}) is not explicitly used.

4.3 Global Term Relevance Weighting

The analysis in Section 4.2 reveals that an important relationship exists between semantic indexing and term weighting. It also shows that the eventual weight w'_i, of a any term t_i in the semantic document representation d' largely depends on the strength of semantic relatedness between t_i and all original terms $t_j \in d$. However, it is also important when computing w'_i to also consider the global importance or relevance of term t_i. It is well known that all terms in a corpus do not have equal importance with some terms having a higher discriminatory power, while many others are not particularly important for distinguishing between classes. However, the resulting weight w_i of any term $t_i \in d'$ from Section 4.2 does not tell us anything about the discriminatory power of t_i. If fact, given any two terms t_1 and t_2 in d', from their respective weights w_1 and w_2 in $\vec{d'}$, there is no way to tell which of the two terms is more relevant for distinguishing between classes and thus, more likely to improve classification performance. In order to capture the importance of terms in d', we introduce a new relevance weight ω_i for t_i that represents the global discriminatory power of term t_i as shown in equation 4.11.

$$d' = (\omega_1 \sum_{j}^{n} w_j Rel(t_1, t_j), \omega_2 \sum_{j}^{n} w_j Rel(t_2, t_j), ..., \ \omega_n \sum_{j}^{n} w_j Rel(t_n, t_j)) \tag{4.11}$$

$$w_i'' = \omega_i \sum_{j}^{n} w_j Rel(t_i, t_j) \tag{4.12}$$

$$d' = (w_1'', w_2'', ..., w_n'') \tag{4.13}$$

Equation 4.11 can be represented in the form of three matrices: a document-term matrix D, semantic-relatedness matrix T, and term-weights matrix W, as shown in Equation 4.14.

$$D' = (D^{rn} \times T^{cn}) \times W \tag{4.14}$$

Where W is a $n \times n$ diagonal matrix and each entry i, j on the leading diagonal (i.e. $i = j$) corresponds to the relevance weight of term $t_i \in V$. Alternatively, the RWSI framework can be viewed as a matrix transformation function H that accepts a conventional term document D and produces a semantic equivalent term document matrix D' as shown in Equation 4.15.

$$H : D \rightarrow D' \tag{4.15}$$

The relevance weight of any term t_i can be estimated using a number of different approaches. However, given the supervised nature of text classification, a good estimate of term relevance can be computed using supervised term weighting approaches. A comprehensive discussion on supervised term weighting was presented in Section 2.4. A basic approach for computing supervised term weights is to use supervised feature selection algorithms. Supervised feature selection provides a statistical score of term importance by looking for informative patterns in the distributions of terms across the different classes in the corpus. Terms whose distributions are more predictive of any one class are assigned a higher weight.

Supervised term weights can be used to populate the leading diagonal of the term weights matrix W. Introducing term weights into Equation 4.14 enables more important terms to have a higher weight in d' which allows them to have a higher influence on document similarity. Many

supervised feature selection techniques have been proposed in the literature. However, Information Gain (IG) and Chi squared (χ^2) have been found to be particularly well suited for text classification (Yang & Pedersen 1997, Forman 2003). Importantly, any effective feature weighting technique can be easily used with the RWSI framework to provide useful term weights.

4.4 Order of Matrix Multiplication

$$D = \begin{array}{c|ccccc} & d_1 & d_2 & d_3 & ... & d_m \\ t_1 & d_{11} & d_{12} & d_{13} & ... & d_{1m} \\ t_2 & d_{21} & d_{22} & d_{23} & ... & d_{2m} \\ t_3 & d_{31} & d_{32} & d_{33} & ... & d_{3m} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ t_n & d_{n1} & d_{n2} & d_{n3} & ... & d_{nm} \end{array}$$

$$T = \begin{array}{c|ccccc} & t_1 & t_2 & t_3 & ... & t_n \\ t_1 & t_{11} & t_{12} & t_{13} & ... & t_{1n} \\ t_1 & t_{21} & t_{22} & t_{23} & ... & t_{2n} \\ t_1 & t_{31} & t_{32} & t_{33} & ... & t_{3n} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ t_1 & t_{n1} & t_{n2} & t_{n3} & ... & t_{nn} \end{array}$$

$$W = \begin{array}{c|ccccc} & t_1 & t_2 & t_3 & ... & t_n \\ t_1 & \omega_{11} & 0 & 0 & ... & 0 \\ t_2 & 0 & \omega_{22} & 0 & ... & 0 \\ t_3 & 0 & 0 & \omega_{33} & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ t_n & 0 & 0 & 0 & ... & \omega_{nn} \end{array}$$

$$D' = \begin{array}{c|ccccc} & d'_1 & d'_2 & d'_3 & ... & d'_m \\ t_1 & d'_{11} & d'_{12} & d'_{13} & ... & d'_{1m} \\ t_2 & d'_{21} & d'_{22} & d'_{23} & ... & d'_{2m} \\ t_3 & d'_{31} & d'_{32} & d'_{33} & ... & d'_{3m} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ t_n & d'_{n1} & d'_{n2} & d'_{n3} & ... & d'_{nm} \end{array}$$

Figure 4.3: Illustration of semantic indexing using RWSI framework.

The order of matrices presented in equation 4.14 is strictly defined. From the properties of

matrices, matrix multiplication is not commutative i.e. $A \times B \neq B \times A$. Thus the order of matrix multiplication presented in equation 4.14 is important. The term relations matrix T is multiplied with the term-document matrix D first, before term weights are introduced using the matrix W. Because matrix multiplication is associative, the same results is obtained by first multiplying the W and T matrices to introduce term weights into term-term relations, and then the result can be multiplied into the term-document matrix D. Changing the order and multiplying the W and D matrices first will lead back to the situation where the final weight of a term is determined by the weights of the terms that it is related to. Consequently, an unimportant term that happens to be related to many important terms can end up with a high weight. We illustrate this situation using the matrices shown in figure 4.3.

Consider the equation $D' = (D \times W) \times T$ where term weights are introduced before terms relations. The entry d'_{11}, which is the weight of term t_1 in document d'_1 in the semantic representation matrix D' is obtained as: $d'_{11} = d_{11}t_{11}\omega_{11} + d_{21}t_{12}\omega_{22} + d_{31}t_{13}\omega_{33} + ...$ Note that the final weight of t_1 in d'_1 is influenced by the relevance weights of all terms in the vocabulary with non-zero relation to t_1. Contrast this with the result of our proposed approach: $d'_{11} = \omega_{11}(d_{11}t_{11} + d_{21}t_{12} + d_{31}t_{13} + ...)$. Note how in this approach, the weight of t_1 in d'_1 is only influenced by the relevance weight of t_1. This ensures that the final weight of any term t_i in d'_j will be proportional to its respective global relevance weight.

4.5 Relevance Weighted Indexing (RWI)

The RWSI framework is not exclusively for semantic indexing. The relevance weighting approach is also effective for supervised document indexing (without semantic relatedness). Recall that in Section 2.4, we described supervised document indexing as the use of supervised term weights for document representation. Thus, it is important to investigate the effect of supervised term weights independently of the influence of semantic relatedness. Semantic indexing can easily be turned off in the RWSI framework by replacing the semantic relatedness matrix T with the identity matrix I as shown in Equation 4.16.

$$D' = (D^{rn} \times I) \times W \tag{4.16}$$

Accordingly, we refer to supervised document indexing using equation 4.16 as Relevance

Weighted Indexing (RWI).

4.6 Evaluation

The aim of our experiments in this section is to evaluate the performance of our RWSI framework for document indexing. Because of the relationship between term weighting and semantic indexing, we decided to evaluate the performance of the RWSI framework separately for binary and *tf-idf* document representations. This allows us to test how the performance of semantic indexing varies between a simpler term weighting approach (binary) and a more complicated term weighting scheme such as *tf-idf*. Accordingly, the evaluation of RWSI on binary representations is presented in Sub-section 4.6.1 and our evaluation on *tf-idf* representations is presented in Sub-section 4.6.2. In both sub-sections, we include in our comparative evaluation baseline BOW representations (no semantic relatedness), semantic representations obtained using the GVSM, and also semantic representations obtained using LSI. For both GVSM and RWSI, we use the document co-occurrence approach (see Section 2.1.2) for computing semantic relatedness. For RWSI, we compute term relevance weights for the matrix W using the Chi squared (χ^2) function.

In Sub-section 4.6.3 we evaluate the performance of using the RWSI framework for supervised document indexing (without semantic relatedness). This allows us to test the utility of the RWI supervised term weighting approach.

All evaluations are performed using standard text classification tasks using a similarity-weighted k Nearest Neighbour (kNN) algorithm where k = 3 and distance is calculated using the cosine similarity metric. Text classification performance is reported using accuracy (see Section 3.1.1). Evaluation is performed using 5-times, 10-fold cross validation with stratification where each fold contains equal number of documents from all classes. Significance is reported at 95% using a standard t-test. For text pre-processing, standard operations of tokenisation and lemmatisation are applied. We also eliminate rare terms (terms with document frequency less than 3). In contrast with Chapter 3, χ^2 feature selection is applied on the vocabulary space of datasets. This allows us to measure the full effect of the terms relevance weighting using the χ^2 function.

4.6.1 Semantic Indexing with binary document vectors

In this part of the evaluation, we test the performance of semantic indexing using the RWSI framework on binary document vectors, which means that each document vector d_i in the document-term matrix D is created using a binary weighting scheme. Accordingly, we compare the following four representations:

- $Base_{bin}$ - Baseline binary document vectors without semantic indexing
- $GVSM_{bin}$ - Semantic indexing with binary document vectors using the GVSM framework (see Section 4.2)
- RWSI$_{bin}$ - Semantic indexing with binary document vectors using our proposed RWSI framework (see Section 4.3)
- LSI_{bin} - Semantic indexing with binary document vectors using LSI (see Section 2.2.2)

Results are presented in Table 4.1 with highest accuracy in each row shown in bold. Values with the '+' sign indicate a statistically significant improvement compared to the baseline $Base_{bin}$ whiles the sign '-' indicates a significantly worse result compared to $Base_{bin}$. Overall results shows that semantic indexing using the RWSI framework (RWSI$_{bin}$) generally performs better than $Base_{bin}$ and $GVSM_{bin}$. The improvements realised using RWSI$_{bin}$ compared to $Base_{bin}$ are statistically significant on 23 out of 37 datasets. On the other hand, the improvements realised using $GVSM_{bin}$ compared to $Base_{bin}$ are significant only on 9 datasets. However, results of $GVSM_{bin}$ are generally better than $Base_{bin}$. In contrast, LSI_{bin} consistently performs worse than $Base_{bin}$. The poor performance of LSI in our evaluation, while unexpected, is not surprising. Similar poor performance of LSI has been previously reported e.g. (Zelikovitz & Hirsh 2001), (Liu et al. 2004), (Kim, Howland & Park 2005), and (Zhang et al. 2008). According to (Zelikovitz & Hirsh 2001) and (Liu et al. 2004), the poor performance of LSI is due to its inability to capture the discriminatory power of terms in document representations. This further confirms our hypothesis that semantic indexing results in a loss of term relevance and that this information is necessary for good text classification performance.

Comparing RWSI$_{bin}$ with $GVSM_{bin}$, RWSI$_{bin}$ performs significantly better on 19 datasets. $GVSM_{bin}$ performs better than RWSI$_{bin}$ on 4 datasets (CryptElectron, ChristianMisc, MarketAd-

Dataset	$\mathbf{Base_{bin}}$	$\mathbf{GVSM_{bin}}$	$\mathrm{RWSI_{bin}}$	$\mathbf{LSI_{bin}}$
Ohsumed				
BactV	81.11	84.53^{+}	$\mathbf{86.86}^{+}$	77.69^{-}
CardR	86.74	88.00	$\mathbf{93.91}^{+}$	83.71^{-}
NervI	86.08	89.81^{+}	$\mathbf{92.04}^{+}$	77.38^{-}
MouthJ	81.83	84.18^{+}	$\mathbf{90.91}^{+}$	79.97^{-}
NeopE	86.58	87.84	$\mathbf{92.39}^{+}$	82.66^{-}
DigNut	85.17	87.98^{+}	$\mathbf{89.34}^{+}$	83.80
MuscS	77.82	82.13^{+}	$\mathbf{87.25}^{+}$	76.09
EndoH	86.80	88.85^{+}	$\mathbf{93.98}^{+}$	83.02^{-}
MaleF	86.99	87.25	$\mathbf{93.89}^{+}$	84.36^{-}
PregN	83.15	86.16^{+}	$\mathbf{88.06}^{+}$	80.99^{-}
ImmunoV	76.00	76.69	$\mathbf{79.51}^{+}$	72.86^{-}
NervM	76.52	82.24^{+}	$\mathbf{86.71}^{+}$	71.91^{-}
RespENT	81.55	84.27^{+}	$\mathbf{88.40}^{+}$	80.44
20 Newsgroups				
HardW	91.1	89.1	$\mathbf{92.9}^{+}$	84.2
MedSp	97.03	97.52	$\mathbf{98.38}^{+}$	92.41^{-}
CryptE	97.73	**97.97**	95.06^{-}	71.39^{-}
ChrisM	93.06	**93.08**	90.86^{-}	81.94^{-}
MeastM	97.66	**97.89**	97.64	89.74^{-}
GunsM	95.46	**95.80**	94.97	84.07^{-}
AutoC	93.39	94.66	$\mathbf{95.08}^{+}$	91.12^{-}
Reuters				
StratM	86.9	85	$\mathbf{88.5}^{+}$	83.0
EntTour	92.7	93.7	90.1	90.7
EqtyB	**94.38**	92.07^{-}	94.17	89.32^{-}
FundA	86.92	85.00^{-}	$\mathbf{89.46}^{+}$	81.60^{-}
InRelD	91.58	90.41	**91.96**	88.81^{-}
NProdRes	83.42	81.50	**84.93**	78.24^{-}
ProdNP	**88.27**	87.56	86.12^{-}	85.28^{-}
OilGas	85.15	83.05^{-}	$\mathbf{88.87}^{+}$	81.85^{-}
ElectGas	85.02	81.16^{-}	$\mathbf{88.19}^{+}$	80.44^{-}
Incident Reports				
Fire	83.80	87.38	$\mathbf{89.45}^{+}$	84.78
Collision	83.05	81.68	**86.10**	81.97
Rollover	80.02	80.55	**80.87**	77.30
CollRoll	85.48	83.80	$\mathbf{90.10}^{+}$	84.02
MiscInc	84.25	**85.60**	85.17	82.28
craneFP	78.3	79.9	76.5	76.4
ShovFP	**84.90**	76.02^{-}	76.83^{-}	74.76^{-}
Movie Reviews				
MovieRev	68.63	70.03	$\mathbf{72.53}^{+}$	63.72^{-}

Table 4.1: Comparison of classification accuracy on different representations using binary vectors.

vert and FinInsurance). The poor performance of RWSI_{bin} on these datasets is likely due to poor weights being learned using χ^2. Perhaps, lack of homogeneity in the documents belonging to the same class in these datasets is responsible for the inability of χ^2 to learn relevant term weights for

these datasets. In general, the results show that semantic indexing is beneficial when used with binary document representations and that our RWSI framework in particular significantly outperforms both traditional BOW representations as well as semantic representation using the GVSM
.

Note that both RWSI and $GVSM$ produce best results on datasets from the Ohsumed corpus. This again shows that the clean language and structure of documents in this corpus makes them very suitable for learning distributional semantic relatedness. $GVSM_{bin}$ produces no significant improvement on any other corpus. However, RWSI$_{bin}$ produces significant improvements on three datasets from the 20Newsgroups corpus, four datasets from the Reuters corpus, two datasets from the Incidents report corpus, as well as on the movie reviews dataset.

4.6.2 Semantic Indexing with *tf-idf* document vectors

In this sub-section, we demonstrate semantic indexing with the RWSI framework, applied to tf-idf document vectors. Accordingly, we compare the following representation schemes:

- $Base_{\textit{tf-idf}}$ - Baseline tf-idf document vectors
- $GVSM_{\textit{tf-idf}}$ - semantic indexing on tf-idf document vectors using document co-occurrence for semantic relatedness (see Section 4.2)
- RWSI$_{\textit{tf-idf}}$ - semantic indexing on tf-idf document vectors using document co-occurrence for semantic relatedness and χ^2 for relevance weighting (see Section 4.3)
- $LSI_{\textit{tf-idf}}$ - Semantic indexing with tf-idf document vectors using LSI (see Section 2.2.2)

Results of the comparative analysis are presented in table 4.2 showing classification accuracy. Best results in each row are presented in bold font. Significant improvements over the baseline ($Base_{\textit{tf-idf}}$) are again presented with '+' sign, while '-' indicates a significantly worse result compared to $Base_{\textit{tf-idf}}$. From the results, we can see that the best classification performance is achieved using RWSI$_{\textit{tf-idf}}$. The performance of RWSI$_{\textit{tf-idf}}$ is significantly better than $Base_{\textit{tf-idf}}$ on 24 out of 37 datasets. RWSI$_{\textit{tf-idf}}$ is significantly better than $GVSM_{\textit{tf-idf}}$ on 31 datasets. Observe that the performance of $GVSM_{\textit{tf-idf}}$ is rather poor. $GVSM_{\textit{tf-idf}}$ is not significantly better than $Base_{\textit{tf-idf}}$ on any dataset and performs significantly worse on 3 datasets.

Dataset	$\mathbf{Base}_{\textit{tf-idf}}$	$\mathbf{GVSM}_{\textit{tf-idf}}$	$\mathrm{RWSI}_{\textit{tf-idf}}$	$\mathbf{LSI}_{\textit{tf-idf}}$
Ohsumed				
BactV	84.50	84.54	$\mathbf{89.60}^{+}$	83.55
CardR	87.22	87.69	$\mathbf{94.48}^{+}$	87.48
NervI	89.06	88.43	$\mathbf{92.71}^{+}$	89.12
MouthJ	86.38	85.03	$\mathbf{91.50}^{+}$	85.64
NeopE	87.71	86.86	$\mathbf{93.83}^{+}$	86.49
DigNut	87.83	87.57	$\mathbf{91.82}^{+}$	88.51
MuscS	82.39	82.43	$\mathbf{89.49}^{+}$	83.21
EndoH	90.31	90.12	$\mathbf{94.50}^{+}$	89.17
MaleF	86.54	85.66	$\mathbf{94.48}^{+}$	86.13
PregN	84.33	84.55	$\mathbf{88.26}^{+}$	84.02
ImmunoV	75.72	74.18	$\mathbf{79.84}^{+}$	76.68
NervM	85.56	85.78	$\mathbf{88.88}^{+}$	85.12
RespENT	83.39	82.82	$\mathbf{89.10}^{+}$	83.42
20 Newsgroups				
HardW	89.7	87.1	92.9	62.6
MedSp	97.87	97.89	$\mathbf{98.84}^{+}$	96.64
CryptE	**97.66**	96.40^{-}	97.18	91.03^{-}
ChrisM	**94.44**	92.71^{-}	91.96^{-}	91.59^{-}
MeastM	**98.34**	97.18^{-}	**98.34**	95.82^{-}
GunsM	**96.20**	95.79	95.30	93.52^{-}
AutoC	96.37	94.61^{-}	$\mathbf{97.65}^{+}$	95.48
Reuters				
StratM	82.9	81.3	$\mathbf{88.6}^{+}$	84.5
EntTour	90.7	90.7	$\mathbf{92.3}^{+}$	91.5
EqtyB	92.48	89.78^{-}	$\mathbf{94.99}^{+}$	92.10
FundA	84.69	81.02^{-}	$\mathbf{89.80}^{+}$	83.80
InRelD	89.18	87.85	$\mathbf{91.97}^{+}$	87.78
NProdRes	78.86	77.86	$\mathbf{82.22}^{+}$	79.31
ProdNP	85.87	84.35	**86.26**	84.26
OilGas	84.75	83.18	$\mathbf{87.64}^{+}$	83.83
ElectG	83.59	82.65	$\mathbf{87.84}^{+}$	83.29
Incident Reports				
Fire0	82.12	81.20	$\mathbf{88.98}^{+}$	80.65
Collision	73.25	75.87	$\mathbf{84.18}^{+}$	70.32
Rollover	77.52	76.97	**77.98**	75.83
CollRoll	82.12	81.08	$\mathbf{85.68}^{+}$	77.32^{-}
MiscInc	80.60	$\mathbf{83.70}^{+}$	81.32^{+}	77.78
CraneFP	78.9	79.6	74.5	78.9
ShovFP	69.94	71.74	**75.44**	67.29
Movie Reviews				
MovieRev	68.08	65.02^{-}	**69.96**	64.57^{-}

Table 4.2: Comparison of classification accuracy on different representations using *tf-idf* vectors.

Note that the poor performance of $GVSM_{\textit{tf-idf}}$ is relative to the performance of $Base_{\textit{tf-idf}}$ and not in absolute terms i.e. the performance of $GVSM_{\textit{tf-idf}}$ is about the same as that of $GVSM_{bin}$ in Table 4.1. This supports our argument in Section 4.3 that unless relevance weights are intro-

duced, information on the global relevance of terms is lost during semantic indexing. $Base_{\textit{tf-idf}}$ generally performs better than $Base_{bin}$ due to the introduction of idf which provides unsupervised relevance weights of terms. However, the benefit from idf is lost during semantic indexing by the GVSM leading to the poor results observed with $GVSM_{\textit{tf-idf}}$. Nonetheless, RWSI$_{\textit{tf-idf}}$ manages to outperform $Base_{\textit{tf-idf}}$ because of the explicit use of term relevance weighting by the RWSI framework.

The performance of LSI on *tf-idf* document vectors is generally much better than LSI_{bin}. This shows that LSI works better on *tf-idf* representation than on binary representation, perhaps because LSI is also able to implicitly take advantage of relevance information from idf. The performance of $LSI_{\textit{tf-idf}}$ is largely comparable to that of $Base_{\textit{tf-idf}}$, with $LSI_{\textit{tf-idf}}$ performing significantly worse than $Base_{\textit{tf-idf}}$ on only 7 datasets. However, no significant gains are achieved using $LSI_{\textit{tf-idf}}$ over $Base_{\textit{tf-idf}}$.

Here also, the group of datasets that performs best with semantic indexing is still the Ohsumed group of datasets. RWSI$_{tfidf}$ produced significant improvements on all 13 datasets in this group. The second group of datasets that benefited most from semantic indexing is the Reuters group with significant improvements from RWSI$_{tfidf}$ on 8 of the 9 datasets in the group. This is double the number compared to RWSI$_{bin}$. On close examination, the significant improvements on these datasets is relative to the poor performance of $Base_{\textit{tf-idf}}$. In other words, the use of idf on these datasets has led to a decline in performance using $Base_{\textit{tf-idf}}$ compared to $Base_{bin}$. However, the decline in performance is not realised with RWSI$_{tfidf}$ which still performs comparable to RWSI$_{bin}$. The implication of this is that semantic indexing using the RWSI framework is able to avoid situations where idf is

In general, comparing the results for *tf-idf* in Table 4.2 with those of binary representation in Table 4.1, $Base_{\textit{tf-idf}}$ is better than $Base_{bin}$ on only 18 out of 37 datasets (48.65%) while $GVSM_{\textit{tf-idf}}$ performs better than $GVSM_{bin}$ on only 9 datasets. This means that both $Base_{bin}$ and $GVSM_{bin}$ perform better than their respective *tf-idf* representations on more than 50% of the datasets. This indicates that for text classification, *tf-idf* is not always a superior weighting scheme compared to binary. In contrast however, RWSI$_{\textit{tf-idf}}$ performs better than RWSI$_{bin}$ on 28 datasets (75.68% of datasets). This indicates that the RWSI framework benefits more from the more complicated *tf-idf* term weighting approach.

Dataset	*tf-idf*	*tf*-CHI	RwI-CHI
Ohsumed			
BactV	84.50	81.11^{-}	$\mathbf{87.35}^{+}$
CardR	87.22	85.20^{-}	$\mathbf{90.80}^{+}$
NervI	89.06	87.13^{-}	**90.16**
MouthJ	86.38	86.94	$\mathbf{89.24}^{+}$
NeopE	87.71	82.42^{-}	$\mathbf{90.81}^{+}$
DigNut	87.83	84.30^{-}	$\mathbf{90.81}^{+}$
MuscS	82.39	84.70^{+}	$\mathbf{87.12}^{+}$
EndoH	90.31	86.88^{-}	$\mathbf{91.93}^{+}$
MaleF	86.54	86.71	$\mathbf{92.40}^{+}$
PregN	84.33	87.28^{+}	$\mathbf{87.36}^{+}$
ImmunoV	75.72	75.34	**77.94**
NervM	**85.56**	82.03^{-}	85.15
RespENT	83.39	82.09	**84.91**
20 Newsgroups			
HardW	89.7	84.1	**90.6**
MedSp	**97.87**	96.31^{-}	97.85
CryptE	**97.66**	92.17^{-}	95.29^{-}
ChrisM	**94.44**	87.15^{-}	91.76^{-}
MeastM	**98.34**	95.49^{-}	96.96^{-}
GunsM	**96.20**	91.36^{-}	94.24^{-}
AutoC	**96.37**	90.10^{-}	95.01^{-}
Reuters			
StratM	81.3	84.4	$\mathbf{86.8}^{+}$
EntTour	90.3	93.3	$\mathbf{95.2}^{+}$
EqtyB	92.48	91.85	$\mathbf{94.39}^{+}$
FundA	84.69	89.00^{+}	$\mathbf{90.54}^{+}$
InRelDef	89.18	$\mathbf{92.84}^{+}$	92.40^{+}
NProdRes	78.86	$\mathbf{84.90}^{+}$	84.23^{+}
ProdNP	85.87	84.36	**86.47**
OilGas	84.75	86.95^{+}	$\mathbf{87.79}^{+}$
ElectGas	83.59	**83.82**	82.86
Incident Reports			
Fire	**82.12**	55.17^{-}	80.40
Collision	73.25	81.90^{+}	$\mathbf{82.22}^{+}$
Rollover	77.52	65.82^{-}	$\mathbf{81.97}^{+}$
CollRoll	82.12	84.78	$\mathbf{86.13}^{+}$
Incidents	80.60	81.03	**81.52**
CraneFP	78.90	81.50	**81.60**
ShovFP	69.94	$\mathbf{81.56}^{+}$	76.93^{+}
Movie Reviews			
MovieRev	68.08	65.78^{-}	**69.35**

Table 4.3: Comparison of supervised indexing approaches against *tf-idf*.

4.6.3 Supervised Indexing

In this sub-section, we compare our RWI supervised term weighting approach with the proposed tf-$\delta(t)$ supervised term weighting approach where idf is replaced with a supervised weighting alternative $\delta(t)$. For both approaches, we use χ^2 for obtaining supervised term weighting. Accordingly we compare the following three representations:

- *tf-idf*- traditional *tf-idf* weighting
- tf-CHI - supervised weighting using tf-χ^2
- RWI-CHI - using the RWSI framework with *tf-idf* document vectors and χ^2 for supervised term weighting, without semantic relatedness.

Results are presented in Table 4.3 where values with the '$^+$' indicate a significant improvement over *tf-idf* performance, and values with '$^-$' indicate a signifcant decline in performance compared to *tf-idf*. Observe from Table 4.3 that the best results are obtained using our proposed RWI-CHI weighting scheme. Specifically, RWI-CHI is better than *tf-idf* on 28 datasets and the improvements on 20 of these datasets are statistically significant. RWI-CHI performs significantly worse than *tf-idf* on only 5 datasets: CryptElectron, ChristianMisc, MideastMisc, GunsMisc, AutoCycle. Note that 4 of these datasets are exactly the same ones that *tf-idf* performed better than $\text{RWSI}_{\textit{tf-idf}}$ which further supports our argument that for these specific datasets, term relevance is not well captured by χ^2. In contrast however, supervised indexing using the tf-CHI approach does not produce consistent improvements compared to *tf-idf*. tf-CHI produces significant improvements only on 8 datasets while it performs worse than *tf-idf* on 16 datasets. Note that this is consistent with the findings of (Debole & Sebastiani 2003) and (Lan et al. 2006) that the tf-$\delta(t)$ supervised approach is often inferior to traditional *tf-idf*. From these results, it is evident that the RWI supervised indexing approach is able to take advantage of the best of both idf and χ^2 for effective term weighting. This also reveals that contrary to previous assumptions (Debole & Sebastiani 2003, Deng et al. 2004, Lan et al. 2006), idf and supervised term weights are complementary and work well together for improved text classification performance.

4.7 Chapter Summary

In this Chapter, we presented a comprehensive analysis of semantic indexing in the VSM. We also provided insights that demonstrate the relationship between semantic indexing and term weighting. We further demonstrated how after semantic indexing, the final weight of a term t_i in the semantic vector of a document d is determined by the number of terms t_i is semantically related to and the strengths of these semantic relationships, regardless of the initial relevance of t_i to d. The implication of this is that local (within-document) importance of terms is lost during semantic indexing. We showed how this can lead to undesired consequences where the weights of less relevant terms are over emphasised by semantic indexing. Consequently, we demonstrated how this problem can be addressed by converting document vectors $\vec{t}$ and semantic relatedness vectors $\vec{\tau}$ into unit vectors using an L2 normalisation function.

We also presented arguments for the need to capture information on the global relevance of terms during semantic indexing. Accordingly, we presented the Relevance Weighted Semantic Indexing (RWSI) framework which introduces term relevance weighting into semantic indexing. We further demonstrated how for text classification, term relevance weights can be learned using supervised feature selection algorithms. We further demonstrated how the RWSI framework can be used for supervised document indexing using the Relevance Weighted Indexing (RWI) approach. We presented a comprehensive evaluation of the RWSI framework using both binary and *tf-idf* document vectors. In both cases, RWSI performs significantly better than both a baseline Bag-Of-Words (BOW) representation with no semantic indexing, as well as semantic indexing using both the GVSM and LSI frameworks. Semantic indexing using GVSM leads to marginal and inconsistent improvements over the baseline. Indeed for *tf-idf* representations, the GVSM hardly made any improvement over BOW representation. This highlights the fact that the global relevance of terms which was captured by idf had been lost during semantic indexing using the GVSM. However, RWSI still produces much significant improvement over baseline *tf-idf*. Thus, an important contribution of this chapter is providing empirical evidence for how the performance of semantic indexing is adversely affected by the inability to capture global term relevance, which is largely responsible for the inconsistent improvements earlier reported.

Finally, we presented a comparative evaluation of supervised indexing using our RWI approach with *tf-idf*, and the popular tf-$\delta(t)$ approach. Results show supervised indexing using our RWSI

framework to significantly outperform both *tf-idf* and tf-$\delta(t)$. The result of using tf-$\delta(t)$ is generally worse than *tf-idf* with only a few improvements. However, the improvements from RWSI are consistent which shows the effectiveness of the RWSI framework for supervised document indexing. Overall, our evaluations show that the best text classification performance is achieved with semantic representations produced using the RWSI framework with *tf-idf* document vectors.

Chapter 5

Supervised Semantic Indexing

Semantic indexing is traditionally an unsupervised process. Accordingly, the semantic indexing approaches we have looked at so far in Chapters 3 and 4 have applied semantic document transformations in an unsupervised manner, ignoring class knowledge in the process. For distributional approaches, semantic relatedness of terms is computed from the entire corpus without particular focus on the class membership of terms. The result of this is that the resulting semantic doc-

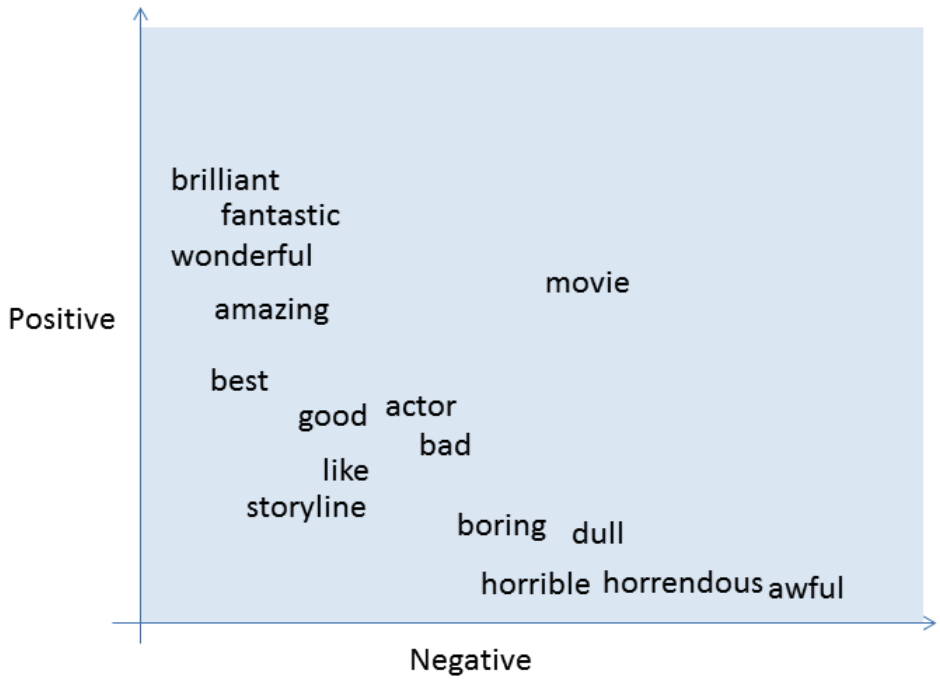

Figure 5.1: Two-dimensional visualisation of terms in the space of *Positive* and *Negative* sentiment classes.

ument representations produced are not likely to be the best fit for the class distribution of the corpus (Bai, Weston, Grangier, Collobert, Sadamasa, Qi, Chapelle & Weinberger 2009, Aggarwal & Zhai 2012).

Consider the example term-document space for a collection of movie reviews shown in Figure 5.1. For the purpose of illustration, terms are shown in the space of *Positive* and *Negative* sentiment classes rather than individual documents. Extracting semantic relatedeness from this term-document space is likely to lead to strong relation between the terms 'good' and 'actor' because of their proximity within the space. This is likely to happen simply because 'actor' is a term that occurs frequently in the corpus and thus, co-occurs often with many other terms in the vocabulary. However, establishing a strong association between 'good' and 'actor' is likely be a source of noise for documents belonging to the *Negative* sentiment class that also happen to contain the term 'actor'. An intuitive approach for addressing this problem is to apply class-specific semantic relatedness values separately to documents belonging to the *Positive* and *Negative* sentiment classes, rather than having a single set of semantic relatedness values for the entire corpus. This way, the term 'actor' is likely to have a weak semantic relation with 'good' in the representation of documents belonging to the *Negative* class, because of the low frequency of occurrence of the term 'good' in that class.

In this chapter, we present a novel approach called Supervised Sub-Spacing ($S3$) for introducing supervision to the semantic indexing process. $S3$ works by creating a separate sub-space for each class within which semantic indexing transformations are applied exclusively to documents that belong to that class. Accordingly, $S3$ requires a separate set of semantic relatedness and term relevance weights to be provided for each class. In this way, $S3$ is able to modify document representations such that documents that belong to the same class are made more similar to one another. In addition, $S3$ is flexible enough to work with a variety of semantic relatedness metrics and yet, powerful enough that it leads to consistent improvements in text classification accuracy, compared to unsupervised semantic indexing.

This chapter is organised as follows, in Section 5.1, we present $S3$ and describe how supervision is introduced into semantic relatedness extraction. The assignment of class-specific term relevance weights is a key step in the $S3$ process. Accordingly, in Section 5.2, we present our approach for learning a class-based term relevance weights. Section 5.3 presents visualisations of a typical term-document space before and after $S3$ transformation, which allows us to demonstrate

how $S3$ brings closer together the representations of documents belonging to the same class. Evaluations of semantic indexing using $S3$ are presented in Section 5.4. We conclude with a chapter summary in Section 5.5.

5.1 Supervised Sub-Spacing

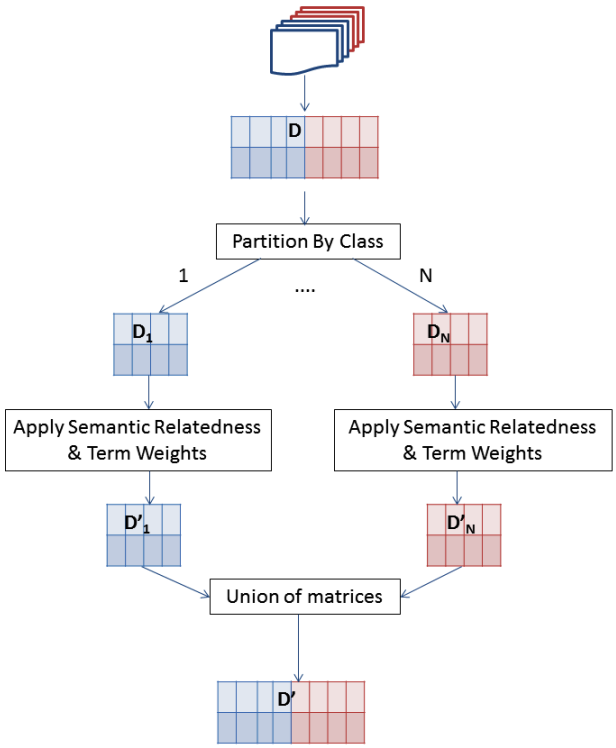

Figure 5.2: Overview of Supervised Sub-Spacing approach to supervised semantic indexing

The primary intuition behind $S3$ is that a separate set of semantic relatedness values and term relevance weights should be computed for terms with respect to each class. Thus the semantic relatedness between any two terms t_i and t_j in class c_k would reflect how semantically close the two terms are in class c_k. Likewise, the weight of any term t_i in class c_k would also indicate how important t_i is with respect to class c_k. To achieve this, we assume that the entire term-document space is composed of N term-document sub-spaces, one for each of N classes in the training corpus. We then apply a transformation function, which consists of assigning semantic relatedness and term weighting, to each sub-space such that documents that belong to the same class are processed together and separate from documents of other classes. Computing semantic

relatedness and term weights in class-partitioned subspaces has the desired effect of making the representations of documents that belong to the same class more similar.

An overview of the $S3$ process is shown in Figure 5.2 where the transformation applied to each subspace is the RWSI function introduced in Chapter 4. Note that while the $S3$ process is not restricted to only binary-class situations, Figure 5.2 highlights only two classes (blue and red) for the purpose to illustration. The term document matrix D, with terms on the rows and documents on the columns, is partitioned by class. Each document vector $d_j \in D$ is expected to belong to at most one class. Semantic transformations are then applied to each class specific sub-space. Finally, a semantic term-document space D' of the same dimensions as the original term-document space D, is created by the union of all document vectors from all individual class-based sub-spaces. Note that this final step is necessary to illustrate that, conceptually, the kNN classifier identifies the k most similar documents by looking at all documents from all classes. However in practice, kNN can be applied separately to each sub-space and then the final ranked list of most similar documents can be composed from the results of the individual sub-spaces.

More formally, a standard term-documents matrix D is initially created from the training corpus where

$$D = \bigcup_{i=1}^{N} D_i = D_1 \cup D_2 \cup ...D_N \tag{5.1}$$

D is an $m \times n$ matrix where m is the total number of documents in the training corpus, n is the number of terms in the indexing vocabulary, and N is total number of classes. Each sub matrix D_i has dimensions $p \times n$ where $p \leq m$ i.e. sub-space D_i contains at most the same number of documents as D and has the same row dimension as D. We define a linear transformation function:

$$H : D_i \rightarrow D'_i \tag{5.2}$$

which transforms each document vector $v \in D_i$ into its semantic representation equivalent $v' \in D'_i$. For the function H, we use our RWSI framework which we introduced in Chapter 4. Thus, details of the linear transformation are as follows.

$$H(D_i) = (D_i^{rn} \times T_i^{cn}) \times W \tag{5.3}$$

Where T_i is an $n \times n$ matrix such that each entry $t_{jk} \in T_i$ represents the strength of the class-specific semantic relatedness between vocabulary terms t_j and t_k. Each entry in T_i is normalised between 0 and 1 with all entries along the leading diagonal (t_{jk} where $j = k$) equal to 1 i.e. the relatedness between any term and itself is 1 (maximum similarity). The semantic term-document space (D') can be constructed from the union of the individual semantic sub-spaces as follows:

$$D' = \bigcup_{i=1}^{N} D'_i = D'_1 \cup D'_2 \cup ...D'_N \tag{5.4}$$

Computing semantic relatedness for each class involves applying any standard semantic relatedness function e.g. document co-occurrence, PMI or LSI (see Section 2.1) on the collection of documents that belong to that class. In this way, a separate set of pair-wise semantic relatedness values are learned with respect to each class c_k, for each term t_i in the indexing vocabulary V .

In Chapter 4 we motivated the need to capture term relevance weights for semantic indexing. However, according to the $S3$ approach, semantic knowledge needed for semantic indexing is provided with respect to each class and not the entire corpus. This means that, unlike in Chapter 4 where term relevance is computed with respect to the entire corpus, for $S3$, a separate set of term relevance weights needs to be calculated with respect to each class. Accordingly, in the next section we describe our approach for computing class-based term relevance weights.

5.2 Class Relevance Term Weighting

The assignment of class-specific relevance term weights for each class is key to the S3 semantic indexing approach. Thus, within the $S3$ framework, any given term t_j can have different weights for different classes $c_k \in C$, each representing the relevance of t_j to that class. It is therefore intuitive to assume that, given a term $t_j \in T$ and candidate class $c_k \in C$, the higher the probability that a document belonging to class c_k contains t_j, the more t_j is considered to be predictive of c_k. This means that the class specific weighting for any term t_j with respect to class c_k can be derived as a function of the probability of observing t_j in a document belonging to class c_k. Accordingly, we can define a simple class relevance weighting (CRW) function as the conditional probability

that a document belonging to the class c_k contains the term t_j as shown in equation 5.5.

$$\text{CRW}(t_j, c_k) = p(d_{c_k}|t_j) \tag{5.5}$$

The conditional probability $p(d_{c_k}|t_j)$ can be decomposed using Bayes' theorem. Recall that in the VSM, a document is simply a set of terms $d_i = \{t_j\}$. Therefore, according to Bayes' theorem, the conditional probability $p(d_{c_k}|t_j)$ can be written as shown in equation 5.6.

$$\text{CRW}(t_j, c_k) = p(d_{c_k}|t_j) = \frac{p(d_{t_j}|c_k)p(c_k)}{p(d_{t_j})} \tag{5.6}$$

Where $p(d_{t_j}|c_k)$ is the conditional probability that a document contains the term t_j given that the document belongs to class c_k and $p(d_{t_j})$ is the probability that any document in the collection contains the term t_j, regardless of the class membership of that document. Both probabilities $p(d_{t_j}|c_k)$ and $p(d_{t_j})$ can be estimated from observed frequency counts in the corpus as shown in equation 5.7.

$$\begin{aligned} p(d_{t_j}|c_k) &= \frac{df(t_j, c_k)}{N_{c_k}} \\ p(d_{t_j}) &= \frac{df(t_j)}{N} \end{aligned} \tag{5.7}$$

Where $df(t_j, c_k)$ is the number of documents that belong to class c_k that contain term t_j, $df(t_j)$ is the number of documents in the entire collection that contain t_j, N_{c_k} is the number of documents that belong to class c_k and N is the number of documents in the entire collection.

One can argue that other functions can equally be applied to learn class-predictive term weights. The first proposal might be to use the probability of the term given the class i.e. $p(t_j|c_k)$. Surely, the higher the conditional probability $p(t_j|c_k)$, the more likely it is that t_j is relevant to c_k. However, one major fault with this argument is that we are assuming higher relevance of the term t_j to the class c_k on the basis of higher document frequency of t_j in c_k. In other words, terms will only have a high weight if they appear in many documents in the class. Given that it is unlikely to have more than a handful of terms appearing in most documents in any given class, using $p(t_j|c_k)$ for term weights will not produce ideal class-predictive term weights.

A second potential term relevance weighting scheme can be adopted from information theory.

Mutual Information measures the mutual dependence between any two given variables. Accordingly, we can derive class-specific weights for any term t_j as the mutual information of t_j with the class c_k as shown in equation 5.8.

$$MI(t_j, c_k) = log_2 \frac{p(t_j, c_k)}{p(t_j)p(c_k)} \tag{5.8}$$

Indeed, equation 5.8 has been widely used as a measure of term-goodness for feature selection. However, note that mutual information is affected by marginal probabilities of terms. This means that MI tends to assign higher weights to rare terms (Yang & Pedersen 1997). MI is also aggressive at assigning zero weight to terms that are not considered to be mutually dependent with the target class. However, this aggressive strategy is not likely to be beneficial for the purpose of assigning class-specific term weights as many of the terms will then be eliminated from indexing. Figure 5.3 shows a comparison of the histograms of term weights derived using our CRW(t_j, c_k) approach with $p(t_j|c_k)$ and $MI(t_j, c_k)$ approaches.

Figure 5.3 shows the distribution of the three different term weighting approaches for 241 distinct terms with respect to the Bacterial class in the BactV dataset with equal distribution of documents in both classes. All weights are normalised between 0 and 1 to allow for comparison between the different term weighting approaches. The x-axis shows bin ranges for the weights in increments of 0.05. Binning is necessary because the weights are continuous values. It is intuitive to assume that each class $c_k \in C$ in a balanced, binary-class corpus will contain a good number of highly relevant terms, a few average terms that are distributed almost equally across both classes, and a large number of low relevance terms that are more relevant to the other class c'_k. Thus, we expect an ideal weighting class-predictive weighting function to reflect this distribution. From figure 5.3, we can see that the distribution of weights learned using our proposed CRW is the one that best reflects the desired distribution of weights.

We further illustrate the difference between the three weighting schemes with the aid of an example. Let t_1, t_2 and t_3 be three terms and c_k be the class for which we wish to calculate class-predictive term weights. Let the sample corpus contain 400 terms, 100 in class c_k and 300 in class $\bar{c_k}$. Let the distribution of terms t_1, t_2 and t_3 in the corpus be as shown in Table 5.1. Term t_1 occurs in 7 documents that belong to class c_k and once in a document that does not belong to c_k. Thus, the numbers shown under the columns c_k and $\bar{c_k}$ are document frequencies of the

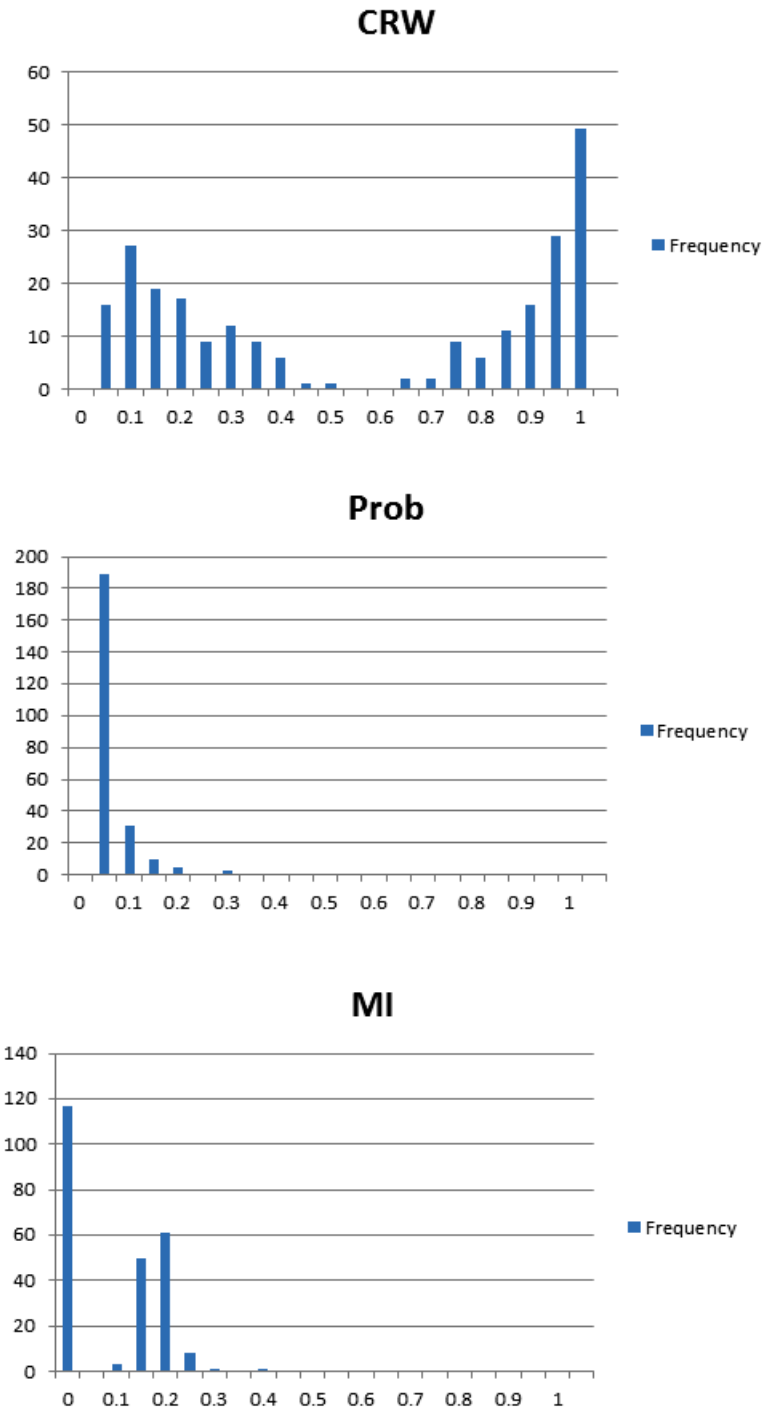

Figure 5.3: Comparison of the histograms of term weights derived using CRW, probabilities (Prob) and Mutual Information (MI).

corresponding row terms within and outside of class c_k respectively. Accordingly, the CRW, Prob and MI weighting of the terms t_1, t_2 and t_3 for class c_k and the complement of c_k are as shown

in Table 5.2. Note that the values of MI have been normalised to between 0 and 1 for the sake of comparison with the other two weighting metrics.

Term	c_k	$\bar{c_k}$
t_1	7	1
t_2	7	6
t_3	30	5

Table 5.1: Distribution of sample terms in the corpus.

Term	c_k			$\bar{c_k}$		
	CRW	Prob	MI	CRW	Prob	MI
t_1	0.875	0.070	0.310	0.125	0.003	0.000
t_2	0.539	0.070	0.190	0.461	0.020	0.000
t_3	0.857	0.300	0.476	0.143	0.017	0.000

Table 5.2: Comparison of term weighting schemes.

Note that terms t_1 and t_3 have a much higher occurrence in documents of class c_k and thus are good predictors of this class. However, this fact is only recognised by the CRW function which assigns a correspondingly high weight to both t_1 and t_3. Prob assigns the same weight to t_1 and t_2 despite the fact that t_2 is not a good predictor of class. This is because Prob. does not utilise information on the occurrence of a term outside of the class of interest. Also, note that none of the terms is assigned a high weight by Prob. which illustrates the likelihood of Prob. to assign low weight to predictive terms. These reasons obviously make Prob. unsuitable for class-predictive term weighting.

MI on the other hand is very sensitive to the occurrence of terms outside of the target class c_k. Note that term t_1 only manages to achieve a weighting of 0.310 despite the fact that t_1 occurs in only a single document outside of c_k. This sensitivity is further highlighted in the case of t_3 which occurs just 5 times outside of c_k, yet MI assigns this a weight of 0.476. This highlights the tendency of MI to downplay the importance of terms that can be considered to be highly predictive of class. Thus, MI is not ideal for learning class-predictive term weights. In contrast, the properties of CRW make it very suitable for class-predictive term weighting.

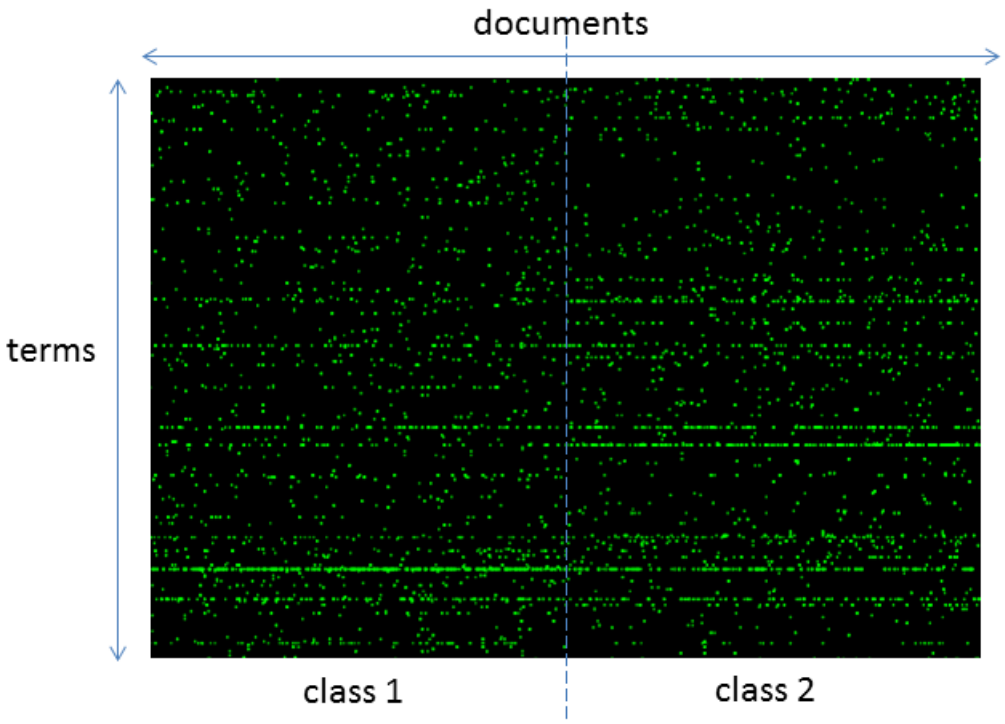

Figure 5.4: Original term-document space.

5.3 Term Space Visualisation

To illustrate the effect of $S3$ on document representations, we present visualisations of the $BactV$ dataset in Figures 5.4 and 5.5. Recall that this is a binary-class dataset created from the Ohsumed corpus, with 500 documents in each class (see Section 2.6). Chi Squared feature selection has also been applied to limit the vocabulary to 300 terms. The column dimensions of Figure 5.4 represent documents while the row dimensions represent terms. Each light coloured point in the space represents a non-zero value, indicating the presence of a term in a document. The dark points are zero-valued indicating the absence of the corresponding term (row) in the corresponding document (column). The space has been organised such that the left half contains documents that belong to the first class and then second half contains documents that belong to the second class. Figure 5.5 shows the same term-document space after semantic indexing using $S3$. Note the difference between the left and right sides of the space is now clearly visible. This indicates how document vectors belonging to the same class have been transformed to be very similar to one another and very different to documents of the other class by incorporating class-specific semantic

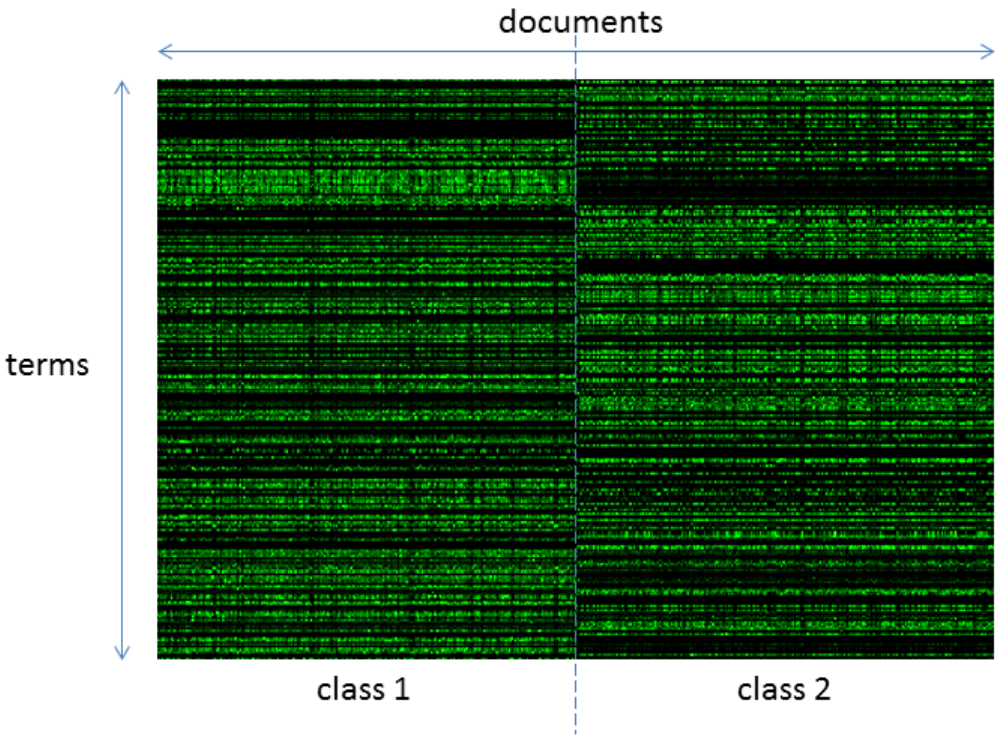

Figure 5.5: Term-document space after $S3$ transformation.

knowledge.

5.4 Evaluation

The aim of this evaluation is three-fold. Firstly, we wish to determine how standard term relatedness metrics are affected by the introduction of supervision using our $S3$ approach. To achieve this we compare classification performance on document representations obtained using the following strategies.

- BASE: Basic BOW representation without semantic indexing
- DOCCOOC: Unsupervised semantic indexing using RWSI with document co-occurrence (DOCCOOC) for semantic relatedness (see Section 2.1.2);
- NPMI: Unsupervised semantic indexing using RWSI with NPMI for semantic relatedness (see Section 2.1.2)

- S3COOC: Supervised semantic indexing using our $S3$ approach with DOCCOOC for semantic relatedness

- S3NPMI: Supervised semantic indexing using our $S3$ approach with NPMI for semantic relatedness

Our expectation is that in comparison with DOCCOOC and NPMI, S3COOC and S3NPMI should lead to better text classification performance. The results for BASE serve as a baseline to measure the improvement achieved using semantic indexing.

The second aim of this evaluation is to study the isolated effect of our probability-based CRW term weighting approach. To achieve this, we create a new representation, $S3_{crw}$, where only class-specific term weights are applied to document representations without semantic relatedness and compare the performance of $S3_{crw}$ with S3COOC and S3NPMI, as well as with BASE. We aim to determine how much of the performance of $S3$ is influenced by the assignment of class-specific relevance weights.

Thirdly, we compare the performance of the two $S3$-based techniques, S3COOC and S3NPMI, to state-of-the-art text classification algorithms. Thus we include a comparison with the following approaches:

- SVM: Basic BOW representation with a Support Vector Machine classifier.

- SPLSI: Supervised semantic indexing using Sprinkled Latent Semantic Indexing approach (see Section 2.3.2) with kNN classifier.

- sLDA: Supervised semantic indexing using supervised Latent Dirichlet Allocation (see Section 2.3.3).

For SVM, we use the libSVM package, for LSI we use the Java Matrix (JAMA) package while for sLDA, we use a freely available C++ implementation [1]. For SVM, and sLDA, we use the default parameter settings of the respective packages. For SPLSI, we use 16 artificial terms per class for sprinkling as described by the authors in (Chakraborti et al. 2006).

Standard preprocessing operations i.e. lemmatisation and stopwords removal are applied to all datasets. For all experiments (except SVM and sLDA), we use a similarity weighted kNN classifier

[1] Available at: http://www.cs.cmu.edu/ chongw/slda/

(with k=3) and using the cosine similarity metric to identify the neighbourhood. Feature selection is also used to limit our indexing vocabulary to the top 300 most informative terms for all datasets except those derived from the incidents report corpus. The documents in these datasets are small in number and their entire vocabulary sizes are generally small so we opted for post feature selection vocabulary size of 100 terms for these datasets. We report classification accuracy averaged over 5 runs of 10-fold cross validation. Statistical significance is reported at 95% using the paired t-test.

5.4.1 Results

Results of comparison between BASE, DOCCOOC, NPMI, S3COOC and S3NPMI are presented in Table 5.3. Values with $^+$ represent a significant improvement over BASE while values with $^-$ represent a significant drop in classification accuracy compared to BASE. Values in the S3COOC and S3NPMI columns that are presented in bold represent a significant improvement over their unsupervised counterparts i.e. DOCCOOC and NPMI respectively. Overall results indicate $S3$-based representations to be significantly superior to their non-supervised counterparts. Comparing DOCCOOC and S3COOC, our $S3$ approach produced an improvement in accuracy on over 89.19% of the datasets and improvements on 75.68% of the datasets are statistically significant. On the other hand, S3NPMI produced better results on 64.86% of datasets compared to NPMI, with improvements on 59.46% of the datasets being statistically significant. Note also that no significant depreciation in performance compared to BASE was observed with any of the $S3$-based representations. Compare this with significant drop in accuracy observed on 4 datasets with DOCCOOC and on 6 datasets with NPMI. This indicates that $S3$ successfully addresses the problem of noisy term relatedness that could harm classification performance.

Consistent with our observations in Chapters 3 and 4, the group of datasets that benefits the most from supervised semantic indexing is the Ohsumed group. On this group of datasets, S3COOC produces significant improvements compared to the DOCCOOC on all 13 datasets, while S3NPMI produces improvements over NPMI on 7. Significant improvements are also realised on the Reuters and Incident Report datasets where S3COOC produced significant improvements on 7 and 4 datasets compared to DOCCOOC respectively. Similarly on these groups of datasets, S3NPMI performs better than NPMI on 8 and 5 datasets respectively. The dataset with the least improvements is the 20Newsgroups dataset. Recall that this is the only dataset with user generated content and most likely contains the highest level of noise in its documents. Accordingly, our eval-

Dataset	BASE	DOCCOOC	NPMI	S3COOC	S3NPMI
Ohsumed					
BactV	85.1	88.6^{+}	90.0^{+}	$\mathbf{90.3}^{+}$	$\mathbf{90.6}^{+}$
CardR	90.0	92.2^{+}	93.8^{+}	$\mathbf{94.3}^{+}$	94.0^{+}
NervI	91.4	91.0	92.9^{+}	$\mathbf{94.0}^{+}$	$\mathbf{93.1}^{+}$
MouthJ	89.9	92.2^{+}	92.9^{+}	$\mathbf{94.0}^{+}$	$\mathbf{94.1}^{+}$
NeopE	91.6	93.8^{+}	94.2^{+}	$\mathbf{95.4}^{+}$	$\mathbf{95.4}^{+}$
DigNut	87.8	91.3^{+}	93.2^{+}	$\mathbf{92.6}^{+}$	93.2^{+}
MuscS	83.1	87.0^{+}	91.1^{+}	$\mathbf{90.9}^{+}$	$\mathbf{91.8}^{+}$
EndoH	91.4	95.8^{+}	96.5^{+}	$\mathbf{96.3}^{+}$	96.7^{+}
MaleF	92.3	94.9^{+}	95.6^{+}	$\mathbf{95.7}^{+}$	95.5^{+}
PregN	89.7	90.4	90.9^{+}	$\mathbf{92.8}^{+}$	$\mathbf{92.2}^{+}$
ImmunoV	78.7	82.5^{+}	84.8^{+}	$\mathbf{85.5}^{+}$	$\mathbf{85.5}^{+}$
NervM	84.5	88.1^{+}	91.0^{+}	$\mathbf{90.0}^{+}$	90.9^{+}
RespENT	87.2	88.1	91.0^{+}	$\mathbf{92.0}^{+}$	93.1
20 Newsgroups					
Hardw	90.1	90.9^{+}	91.3^{+}	$\mathbf{92.5}^{+}$	$\mathbf{92.64}^{+}$
MedSp	95.9	93.4^{-}	95.8	**95.6**	95.2
CryptE	96.3	90.3^{-}	91.8^{-}	**96.0**	**95.4**
ChrisM	88.9	90.5^{+}	89.9^{+}	90.8^{+}	88.9^{+}
MeastM	95.6	95.3	94.9	95.8	94.7
GunsM	93.7	94.0	94.0	94.1	93.9
AutoC	93.7	95.1	96.2^{+}	$\mathbf{95.8}^{+}$	96.2^{+}
Reuters					
StratM	88.5	89.4	83.7^{-}	$\mathbf{92.0}^{+}$	$\mathbf{91.4}^{+}$
EntTour	94.3	95.7^{+}	95.3	95.2^{+}	94.3
EqtyB	95.5	95.5	94.8^{-}	$\mathbf{95.9}^{+}$	$\mathbf{95.9}^{+}$
FundA	89.4	92.0^{+}	89.9	$\mathbf{92.6}^{+}$	$\mathbf{91.5}^{+}$
InRelD	92.3	94.1^{+}	91.7	94.2^{+}	$\mathbf{93.9}^{+}$
NProdRes	85.5	86.9	80.4^{-}	$\mathbf{89.6}^{+}$	$\mathbf{86.5}^{+}$
ProdNP	87.7	89.3^{+}	88.4	$\mathbf{90.2}^{+}$	$\mathbf{89.9}^{+}$
OilGas	87.3	86.3^{-}	85.7^{-}	**88.1**	**87.7**
ElectG	88.7	84.6^{-}	84.0^{-}	**87.1**	**88.3**
Incident Reports					
Fire	87.3	93.4^{+}	92.3^{+}	92.7^{+}	$\mathbf{94.1}^{+}$
Collision	88.6	91.2^{+}	93.3^{+}	$\mathbf{93.9}^{+}$	$\mathbf{95.7}^{+}$
Rollover	86.1	89.5^{+}	90.7^{+}	$\mathbf{92.2}^{+}$	$\mathbf{92.2}^{+}$
CollRoll	90.6	93.9^{+}	93.4^{+}	$\mathbf{96.1}^{+}$	$\mathbf{95.5}^{+}$
MiscInc	81.5	84.4^{+}	89.8^{+}	$\mathbf{88.7}^{+}$	$\mathbf{90.4}^{+}$
CraneFP	93.8	94.6	95.4	94.7	95.5^{+}
ShovFP	94.1	95.4^{+}	96.2^{+}	95.4	96.0^{+}
Movie Reviews					
MovieRev	70.7	78.8	82.2	$\mathbf{83.4}^{+}$	$\mathbf{85.0}^{+}$

Table 5.3: Comparison of supervised and unsupervised term relatedness on binary classification tasks.

uation shows that clean documents are important for effective distributional semantic relatedness extraction.

Dataset	BASE	DOCCOOC	NPMI	S3COOC	S3NPMI
Science	80.8	77.9	73.2	$\mathbf{82.6}^{+}$	$\mathbf{83.0}^{+}$
Ohsumed01	52.0	51.7	52.5	$\mathbf{56.7}^{+}$	$\mathbf{58.0}^{+}$
Ohsumed02	45.2	44.3	43.0	$\mathbf{55.0}^{+}$	$\mathbf{55.6}^{+}$
Ohsumed03	47.8	50.2	50.0	$\mathbf{58.4}^{+}$	$\mathbf{56.1}^{+}$
Ohsumed04	31.9	33.5	32.8	$\mathbf{40.6}^{+}$	$\mathbf{39.2}^{+}$

Table 5.4: Comparison of supervised and unsupervised term relatedness on multi-class classification tasks.

Table 5.4 compares between BASE, DOCCOOC, NPMI, S3COOC and S3NPMI on multi-class classification tasks. Note that the results are consistent with that of binary classification. Both S3COOC and S3NPMI significantly outperform the unsupervised approaches, NPMI and S3COOC. Also note the Science and Ohsumed02 datasets where the performance of NPMI and DOCCOOC is worse that BASE. Again, the use of supervision by S3COOC and S3NPMI produces significant improvements compared to BASE which further supports that supervision addresses the problem of noise associated with unsupervised semantic relatedness.

5.4.2 S3 for Supervised Term Weighting

In this section we evaluate the performance of $S3$ for supervised term weighting. Given the importance of class relevance term-weighting to $S3$, it is important to study the isolated effect of the class relevance term weighting without semantic relatedness. This also allows us to determine the effectiveness of $S3$ and the CRW approach for supervised term weighting. Table 5.5 compares the results obtained with $S3_{crw}$ which is $S3$ with class relevance weighting only (without semantic relatedness) with the performance of BASE, S3COOC and S3NPMI where values with $^{+}$ indicate significant improvement over BASE. Significant improvements are achieved using $S3_{crw}$ on 51.35% of the datasets compared to BASE. This shows that $S3$ is effective for supervised term weighting even in the absence of semantic relatedness. However, the improvement achieved using $S3_{crw}$ is not as substantial as that achieved using $S3$-based semantic representations (S3COOC and S3NPMI) where significant improvement is achieved, compared to BASE, on over 70% of the datasets. This shows that the combination of semantic relatedness and class relevance weighting using $S3$ produces the best improvements in text classification performance.

Dataset	BASE	S3$_{crw}$	S3COOC	S3NPMI
Ohsumed				
BactV	85.1	85.6	90.3^{+}	90.6^{+}
CardR	90.0	91.8^{+}	94.3^{+}	94.0^{+}
NervI	91.4	91.9	94.0^{+}	93.1^{+}
MouthJ	89.9	91.0^{+}	94.0^{+}	94.1^{+}
NeopE	91.6	93.7^{+}	95.4^{+}	95.4^{+}
DigNut	87.8	89.9^{+}	92.6^{+}	93.2^{+}
MuscS	83.1	85.5^{+}	90.9^{+}	91.8^{+}
EndoH	91.4	93.3^{+}	96.3^{+}	96.7^{+}
MaleF	92.3	93.6^{+}	95.7^{+}	95.5^{+}
PregN	89.7	90.4^{+}	92.8^{+}	92.2^{+}
ImmunoV	78.7	80.5^{+}	85.5^{+}	85.5^{+}
NervM	84.5	85.2^{+}	90.0^{+}	90.9^{+}
RespENT	87.2	89.9^{+}	92.0^{+}	93.1
20 Newsgroups				
Hardw	90.1	90.9^{+}	92.5^{+}	92.64^{+}
MedSp	95.9	95.6	95.6	95.2
CryptE	96.3	95.0	96.0	95.4
ChrisM	88.9	90.1	90.8^{+}	88.9^{+}
MeastM	95.6	94.6	95.8	94.7
GunsM	93.7	93.0	94.1	93.9
AutoC	93.7	94.1	95.8^{+}	96.2^{+}
Reuters				
StratM	88.5	90.7	92.0^{+}	91.4^{+}
EntTour	94.3	94.6	95.2^{+}	94.3
EqtyB	95.5	96.0^{+}	95.9^{+}	95.9^{+}
FundA	89.4	90.9^{+}	92.6^{+}	91.5^{+}
InRelD	92.3	94.1^{+}	94.2^{+}	93.9^{+}
NProdRes	85.5	88.3^{+}	89.6^{+}	86.5^{+}
ProdNP	87.7	88.2	90.2^{+}	89.9^{+}
OilGas	87.3	88.8^{+}	88.1	87.7
ElectG	88.7	89.5^{+}	87.1	88.3
Incident Reports				
Fire	87.3	87.3	92.7^{+}	94.1^{+}
Collision	88.6	89.6	93.9^{+}	95.7^{+}
Rollover	86.1	89.0^{+}	92.2^{+}	92.2^{+}
CollRoll	90.6	92.1	96.1^{+}	95.5^{+}
MiscInc	81.5	82.4	88.7^{+}	90.4^{+}
CraneFP	93.8	93.0	94.7	95.5^{+}
ShovFP	94.1	93.3	95.4	96.0^{+}
Movie Reviews				
MovieRev	70.7	71.1	83.4^{+}	85.0^{+}

Table 5.5: Comparison of term-weighting only with $S3$.

Dataset	SVM	SPLSI	sLDA	S3COOC	S3NPMI
Ohsumed					
BactV	90.2	88.6	89.3	90.3	**90.6**
CardR	93.7	93.7	92.76	**94.3**	94.0
NervI	92.2	90.3	91.9	**94.0**	93.1
MouthJ	91.8	93.4	92.3	**94.0**	**94.1**
NeopE	93.5	94.5	94.8	**95.4**	**95.4**
DigNut	91.6	90.7	91.5	92.6	**93.2**
MuscS	89.8	89.7	89.2	90.9	**91.8**
EndoH	94.0	95.4	93.7	96.3	**96.7**
MaleF	94.4	94.7	92.9	**95.7**	95.5
PregN	89.6	91.9	89.6	**92.8**	91.4
ImmunoV	82.4	83.3	81.0	**85.5**	83.6
NervM	88.3	90.0	87.7	90.0	**90.9**
RespENT	90.5	92.0	90.2	92.0	**93.1**
20 Newsgroups					
Hardw	92.4	92.9	91.3	92.5	92.64
MedSp	**97.1**	95.3	95.7	95.6	95.2
CryptE	**96.9**	89.1	93.7	**96.0**	95.4
ChrisM	90.8	90.6	**91.7**	90.8	88.9
MeastM	95.7	93.2	95.0	**95.8**	94.7
GunsM	92.2	93.5	92.68	**94.1**	93.9
AutoC	95.9	95.6	**97.0**	95.8	96.2
Reuters					
StratM	89.7	**92.7**	91.1	92.0	91.4
EntTour	**96.0**	94.7	93.6	95.2	94.3
EqtyB	**96.1**	96.0	95.2	95.9	95.9
FundA	90.9	91.3	**93.1**	92.6	91.5
InRelD	92.0	93.4	**94.9**	94.2	93.9
NProdRes	85.2	85.8	87.7	**89.6**	86.5
ProdNP	86.4	89.3	87.8	**90.2**	**89.9**
OilGas	**88.8**	86.6	88.6	88.1	87.7
ElectG	90.6	89.2	**93.02**	87.1	88.3
Incident Reports					
Fire	91.9	92.3	46.4	92.7	**94.1**
Collision	89.7	95.5	46.2	93.9	**95.7**
Rollover	91.5	89.8	50.2	**92.2**	**92.2**
CollRoll	93.8	96.2	50.8	**96.1**	95.5
MiscInc	**92.5**	89.1	50.6	88.7	90.4
CraneFP	94.6	95.2	45.1	94.7	95.5
ShovFP	**97.7**	95.1	43.9	95.4	96.0
Movie Reviews					
MovieRev	80.1	75.7	81.7	83.4	**85.0**

Table 5.6: Comparison of $S3$ techniques with SVM, SPLSI and sLDA

5.4.3 Comparison with state-of-the-art

Table 5.6 compares the results our two $S3$ approaches with those of SVM, Sprinkled LSI (SPLSI) and supervised LDA (sLDA). Values in bold represent the best results in each row. The overall

significant improvement over SVM indicates a clear advantage from $S3$-based representations for text classification. For instance, S3COOC is better than SVM on 69.44% of the datasets, (significantly on 51.35% of the datasets) while S3NPMI is better than SVM on 67.56% (significantly on 48.64%). In comparison with SPLSI, S3COOC performs better on 75.67% of the datasets (significantly on 67.56%). On the other hand, S3NPMI outperforms SPLSI on 70.27% of the datasets with significant improvements also on 62.16%.

Comparing S3COOC with sLDA, S3COOC is better on 78.38% of datasets (significantly on 64.86%). In contrast, sLDA is significatly better than S3COOC on only 2 datasets: AutoCycle and ElectGas. Observe that sLDA performs particularly poorly on the incident report datasets, Fire, Collision, Rollover, CollRoll, MiscInc and ShovFP. These datasets have a total of only 200 documents (100 documents per class). This indicates that perhaps the number of documents in these datasets is too small for sLDA to learn accurate supervised topic models. Note that accuracy is about 50% for these datasets (about 46% for Fire and Collision). However, S3COOC and S3NPMI produce the best accuracies on these datasets except on MiscInc and ShovFP where SVM performs best. This shows that semantic indexing with $S3$ is effective on both large and small datasets.

5.5 Chapter Summary

In this chapter, we have introduced a novel technique called Supervised Sub-Spacing ($S3$) for introducing supervision into semantic indexing. We presented a detailed evaluation of this approach on 36 datasets from a variety of different domains including news stories, medical abstracts and incident reports. We investigated $S3$ with two semantic relatedness metrics: document co-occurrence (DOCCOOC) and Normalised Point-wise Mutual Information (NPMI). Results show $S3$ leads to improvements in the performance of these two metrics on over 80% of the datasets. We also compared two $S3$-based approaches (S3COOC and S3NPMI) with SVM, a supervised version of Latent Semantic Indexing (SPLSI) that uses a technique called Sprinkling, and supervised LDA (sLDA). Results show that our $S3$-based approaches outperform SVM, SPLSI and sLDA on over 70% of datasets.

The effectiveness of $S3$ lies in its ability to transform document representations such that documents that belong to the same class are made more similar to one another while, at the same time, making them more dissimilar to documents of a different class. We presented visualisations

of a typical term-document space before and after $S3$ transformation in order to demonstrate the effect of $S3$ on document representations. We also showed how supervised term weighting using the class relevance term weighting (CRW) approach contributes to improved text classification performance.

The $S3$ technique we presented here has a number of additional advantages compared to other supervised semantic indexing approaches. Firstly, unlike sLDA and SPLSI, $S3$ is not tied to any specific semantic relatedness approach (i.e. LDA with SLDA, and LSI with SPLSI). We demonstrated this by using $S3$ with both DOCCOOC and NPMI semantic relatedness approaches. Secondly, unlike sprinkling, $S3$ does not require higher order semantic relatedness. This means that $S3$ does not apply restrictions to the type of semantic relatedness metric that can be used. A third advantage is that $S3$ does not require any parameter tuning whereas sprinkling requires a predetermined number k of artificial terms to be injected into the vocabulary while sLDA requires the optimum number of topics to be determined. In both cases, it is unlikely that globally optimum parameter settings exists and thus, the optimum number of sprinkled terms as well as the optimum number of topics will have to be determined individually for each dataset which further contributes to the complexity of these approaches.

Chapter 6

Case Study: Sentiment Classification using $S3$

Sentiment classification is the task of assigning opinion documents to the categories "Positive" and "Negative" in order to indicate the type of opinion or sentiment expressed in the documents (Liu 2010). Sentiment classification can essentially be modelled as a binary classification task, which means that techniques used for traditional text classification are equally applicable for classifying opinion documents. To train a machine learning classifier, a collection of opinion documents with known sentiment labels is used. A common document representation approach for sentiment classification is also a standard VSM where all terms in the opinion documents are used as features (Pang et al. 2002). Given any new opinion document d_q with unknown sentiment class, the classifier is then used to predict the appropriate sentiment category to assign d_q. Different machine learning algorithms have been successfully employed for sentiment classification e.g. SVM, Naive Bayes and Maximum Entropy (Pang et al. 2002) with typically high sentiment classification accuracy (Muhammad, Wiratunga, Lothian & Glassey 2013).

Despite the success of machine learning for sentiment classification, recent works indicate that machine learning approaches can benefit from using background knowledge from sentiment lexicons (Melville, Gryc & Lawrence 2009, Dang, Zhang & Chen 2010, Mudinas et al. 2012). Combining the two approaches has a number of benefits. Firstly, it allows machine learning classifiers to utilise general knowledge relevant for sentiment classification, thus, avoiding overfitting the training data. Secondly, supplementing training data with knowledge from sentiment lexicons

has the potential to reduce the number of training examples required to build accurate classifiers. However, achieving significant improvements using this combined approach has proved difficult (Mudinas et al. 2012).

In this chapter, we present a case study of applying our $S3$ approach (see Chapter 5) to the task of sentiment classification. We also demonstrate how background knowledge from a sentiment lexicon can be utilised with the $S3$ approach for improved sentiment classification performance. This Chapter is organised as follows: In Section 6.1 we describe the application of $S3$ to the task of sentiment classification. Section 6.2, describes how the SentiWordNet lexicon is used to provide sentiment scores of terms which are then utilised for semantic indexing of subjective text using $S3$. We present the datasets we use for evaluation in Section 6.3. Evaluation of $S3$ on sentiment classification tasks is presented in Section 6.4. We conclude the chapter with a summary in Section 6.5

6.1 $S3$ for Sentiment Classification

Given the similarity that exists between sentiment classification and standard text classification, we expect similar improvements which were achieved using semantic indexing on text classification to be achieved on sentiment classification. Hence, the aim of this chapter is to apply semantic indexing using the $S3$ approach to the task of sentiment classification. An overview of $S3$ applied to sentiment classification is presented in Figure 6.1.

Recall that in sentiment classification, the objective is to classify opinion documents into 'Positive' and 'Negative' sentiment classes. Thus, sentiment classification is essentially a binary classification task involving these two classes. Accordingly, applying $S3$ for semantic indexing of opinion documents involves partitioning the term-document space D into 'Positive' and 'Negative' classes (D_{pos} and D_{neg} respectively) and then learning semantic relations (T_{pos} and T_{neg}) and class relevance weights (W_{pos} and W_{neg}) separately for 'Positive' and 'Negative' documents. This way, semantic relatedness between positive opinion terms is emphasised within the representations of positive documents. Similar emphasis is also applied to negative terms within negative document representations. Document transformation is then applied to the class-specific term-document spaces (D_{pos} and D_{neg}) to produce the semantic term-document spaces D'_{pos} and D'_{neg} respectively. The final semantic term-document space D' is constructed as a union of D'_{pos} and

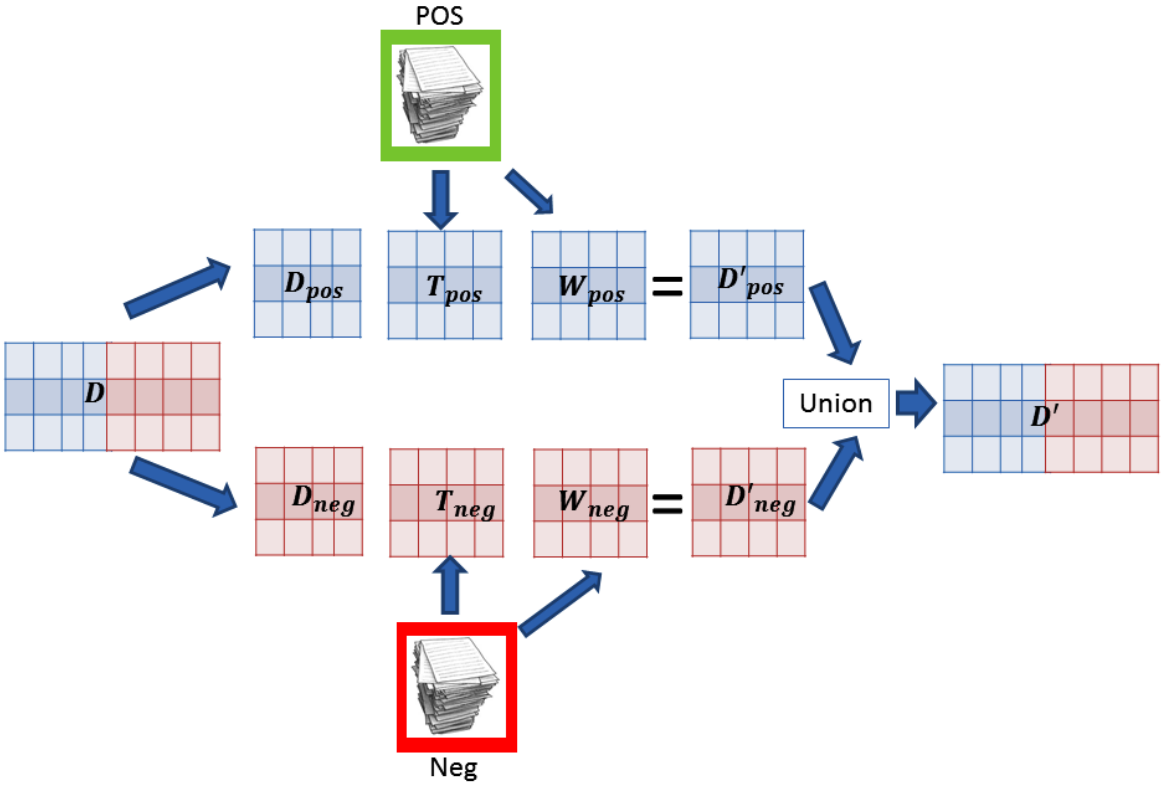

Figure 6.1: Semantic indexing for sentiment classification using $S3$

D'_{neg}.

The effect of $S3$ on opinion documents is that the representations of positive documents are brought closer together in the vector space and are made more distant from the representations of negative documents. This results in a more linearly separable term-document space which in turn should improve sentiment classification performance. The partitioning of opinion documents into subspaces by $S3$ also provides opportunity for utilising additional class-specific knowledge to further improve document representation, as we will discuss in the next section.

6.2 Combining $S3$ with SentiWordNet

So far, we have demonstrated how class relevance weights can be learned directly from the training corpus. However, for sentiment classification, the relevance of a term to a sentiment category can be learned from sources other than corpus statistics e.g. sentiment lexicons. A sentiment lexicon is a collection of opinion terms together with an indication of the sentiment that these terms convey.

Typically, a sentiment term is associated with a numerical value along each sentiment dimension (Positive and Negative) in the lexicon, indicating the strength of the opinion associated with that term along that dimension. For example, the term "excellent" could be associated with the positive score *0.9* and negative score *0.1* out of a maximum possible score of 1.0, indicating that "excellent" is strong indicator of positive sentiment and a weak indicator of negative sentiment. Approaches that use sentiment lexicons for sentiment classification typically make a classification decision using the scores returned by the sentiment lexicon for all terms in a document.

Our goal here is to utilise sentiment scores from a sentiment lexicon to improve semantic indexing of opinion documents using $S3$. Note that, similar to the class relevance term weights, sentiment scores from lexicons also provide the degree of relevance of a sentiment term to a sentiment class. Thus, we aim to use sentiment scores from a sentiment lexicon as further evidence for the relevance of sentiment terms by combining with the class relevance weights extracted from the corpus. Accordingly, we wish to obtain a new weight $w(t_i, c_j)$ for a term t_i by augmenting the class relevance weight $CRW(t_i, c_j)$ of t_i extracted from corpus statistics, with the class-specific sentiment score ($score(t_i, c_j)$) of t_i obtained from a sentiment lexicon as shown in equation 6.1.

$$w(t_i, c_j) = \alpha CRW(t_i, c_j) + (1 - \alpha) score(t_i, c_j) \tag{6.1}$$

Where both $CRW(t_i, c_j)$ and $score(t_i, c_j)$ are normalised within the range 0 and 1. The value α is used to control the contribution from the class relevance weight and that from the sentiment lexicon to the final weight $w(t_i, c_j)$. For the purpose of this work, we use the value $\alpha = 0.5$.

We decide to use of a combination of both CRW and sentiment score because, we view the two as being complementary. Indeed, while sentiment lexicons are certainly useful, they have been found to not be sufficient for sentiment classification for a number of reasons (Liu 2012). Firstly, the context within which a term is used is very important for accurately determining its sentiment. However, context is not available to sentiment lexicons, and this needs to be captured directly from the documents. A second reason why using a combined approach is better is the problem of lexicon coverage. Sentiment lexicons are only able to provide scores for terms that exist in their dictionary which means that sentiment scores will not be available for terms that exist outside of the dictionaries of the sentiment lexicon. However, by combining with CRW, we enure that the scores of terms that may be absent from the lexicon are still captured.

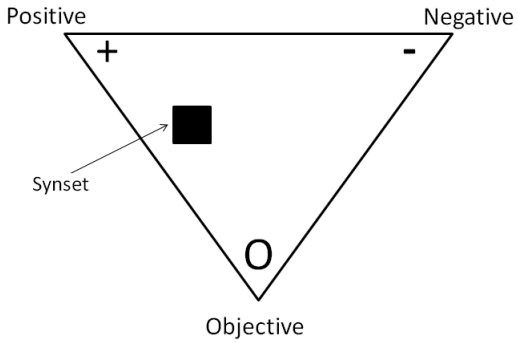

Figure 6.2: Representation of the position of a synset in three-dimensional sentiment space as provided by SentiWordNet.

For the purpose of this work, we use SentiWordNet (Baccianella & Sebastiani 2010) which is a high coverage sentiment lexicon developed as an extension to the popular WordNet lexical resource. Accordingly, SentiWordNet has very much the same structure as WordNet with terms grouped together into synonym sets called synsets or concepts (we use the words synset and concept to denote the same thing). Synsets are further assigned into one of Noun, Verb, Adjective and Adverb dictionaries based on their part-of-speech category. Each synset in SetiWordNet is associated with scores along three sentiment dimensions, a negative score, a positive score and an objective score, indicating how strongly that entry is associated with the respective sentiment dimension. The positive, negative and objective scores of each entry sum to a total of 1.0. An alternative way of visualising this is in a three dimensional sentiment space where a synset can be considered as occupying a position in this space as show in Figure 6.2.

Given any lemmatised term t_i, we obtain its sentiment score from SentiWordNet by matching t_i the appropriate synset in SentiWordNet. Terms are matched to synsets by searching for matching entries in the Noun, Verb, Adverb and Adjective dictionaries in that order. The order used for dictionary lookup corresponds to the order of size of the dictionaries i.e. the dictionary with the most number of entries is the Noun dictionary followed by the Verb dictionary etc. If a matching entry is found in any dictionary then the lookup is abandoned and subsequent dictionaries are not searched. Our decision not to use part-of-speech tagging means that our approach is not limited by the accuracy of a part-of-speech tagger. Also, many part-of-speech tagger use a more expansive set of part-of-speech categories than the four categories used by SentiWordNet. This means that a mapping is required from the part-of-speech label assigned by the tagger to the appropriate

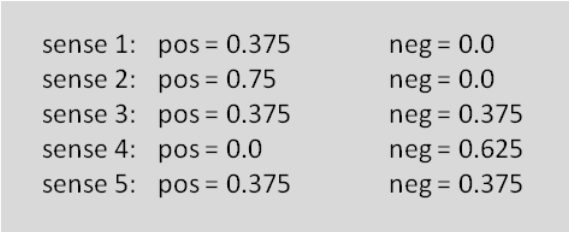

Figure 6.3: Matching synsets for the term 'fantastic' in SentiWordNet showing both negative and positive sentiment scores for each sense

part-of-speech dictionary in SentiWordNet.

The final sentiment score of term t_i is obtained as the average score of all matching synsets in the target dictionary, along the *positive* and *negative* sentiment dimensions. For example the term 'fantastic' matches 5 synsets in the Noun dictionary as shown in Figure 6.3. Thus, the score of 'fantastic' for the Positive class is obtained as the average of the positive scores of all 5 senses. The same approach is used for the Negative class.

Once the class-specific sentiment score $score(t_i, c_j)$ of t_i has been obtained, $w(t_i, c_j)'$ is computed by combining $score(t_i, c_j)$ and $w(t_i, c_j)$ using a linear interpolation approach as shown in equation 6.1.

6.3 Datasets

A summary of the datasets used in our evaluation is provided in Table 6.1 which shows the names of each dataset, the number of documents and the average vocabulary size, which is the average number of unique terms in each document. All datasets contain only the binary sentiment classes *Postive* and *Negative* with equal distribution of documents between the two classes. We describe these datasets in detail in the following sub-sections.

Dataset	Number of documents	Ave doc. Vocabulary Size
Movie Reviews	1000	197.4
Amazon Reviews	1000	18.0
Twitter Dataset	900	5.9
Hotel Reviews	1000	48.8

Table 6.1: Overview of datasets used for evaluation showing number of documents in each dataset.

6.3.1 Movie Reviews

This is a sentiment classification corpus comprising movie reviews from the Internet Movie Database (IMDB) (Pang et al. 2002). We used version 1 of this corpus which contains 1400 reviews, half of which are classified as expressing positive sentiment while the other half is classified as negative. Accordingly, the classification task for this dataset is to determine the sentiment orientation of any given review.

6.3.2 Amazon Reviews

This is another sentiment classification corpus consisting of customer reviews obtained from the Amazon website. We used version 1 of this dataset which is described in (Blitzer, Dredze & Pereira 2007). Four types of products were considered in the dataset: books, DVDs, electronics and kitchen appliances. The original user reviews had a star rating between 1 and 5. We transformed this into binary sentiment classes using the same approach as (Blitzer et al. 2007) where reviews with star rating less than 3 are considered negative and those with star rating of 4 and 5 are considered positive.

6.3.3 Twitter Dataset

This is a collection of 5513 tweets on four topics: *Apple*, *Google*, *Microsoft* and *Twitter*, available from Sanders Analytics [1]. All tweets have been manually classified into one of three sentiment categories: negative, positive, neutral, including an additional uncategorised category for tweets that are not considered to bear any sentiment. We utilise only the positive and negative sentiment classes for our evaluation.

6.3.4 Hotel Reviews

This is a collection of hotel reviews obtained from the TripAdvisor website as described in (Wang, Lu & Zhai 2010). The corpus contains a total of 235,793 reviews, each with a user assigned star rating between 1 and 5. We convert these ratings into binary sentiment classes by labeling reviews with a star rating lower than 3 as negative while reviews with a rating above 3 are tagged

[1] http://www.sananalytics.com/lab/twitter-sentiment/

as positive. We then randomly select 500 reviews from each of the positive and negative classes to create our evaluation dataset.

We took subsamples of the original corpora to create our datasets for the sake of computational efficiency.

6.4 Evaluation

The aim of our evaluation is two-fold. Firstly, we wish to determine the performance of semantic indexing using $S3$ on sentiment classification. Secondly, we wish to evaluate the performance of extending $S3$ with sentiment scores from SentiWordNet. To achieve this we compare sentiment classification performance on document representations obtained using the following strategies.

- BASE: Basic BOW approach without term relatedness
- S3COOC: Supervised term-relatedness extracted using our $S3$ approach with DOCCOOC term-relations (see Section 5.1)
- S3NPMI: Supervised term-relatedness extracted using our $S3$ approach with NPMI term-relations (see Section 5.1)
- S3COOCSWN: S3COOC augmented with SWN sentiment scores (see Section 6.2)
- S3NPMISWN: S3NPMI augmented with SWN sentiment scores (see Section 6.2)

We apply standard text pre-processing steps of stopwords removal and lemmatisation. We eliminate terms with a document frequency of less than 3. We then use Chi squared feature selection to limit the vocabulary to the top 300 terms for each dataset. Classification accuracy is reported using a similarity weighted kNN classifier (with k=3) and using the cosine similarity metric to identify the neighbourhood. Our expectation is that semantic indexing using S3COOC and S3NPMI will produce better results on sentiment classification compared to non semantic representation (BASE) because of $S3$'s ability to produce semantic document representations that are a better fit for the underlying class distribution. We also expect that the use of sentiment scores in S3COOCSWN and S3NPMISWN should lead to even better sentiment classification performance compared to S3COOC and S3NPMI.

Dataset	BASE	S3COOC	S3NPMI
MovieReviews	70.7	83.4+	**85.0**+
AmazonReviews	65.9	78.7+	**81.3**+
TwitterData	71.6	**82.9**+	82.7+
HotelReviews	64.5	**68.4**+	67.3+

Table 6.2: Results of semantic indexing using two $S3$-based representation.

Table 6.2 shows the results of comparing the standard $S3$ semantic representations with BASE (baseline representation without semantic indexing). The results for BASE serve as a baseline to measure the improvement achieved using semantic indexing. Best results for each dataset are shown in bold. Values with the + sign indicate a statistically significant improvement compared with BASE. Observe that semantic indexing using $S3$ leads to statistically significant improvements on all datasets. S3NPMI outperforms S3COOC on the MovieReviews and AmazonReviews dataset while S3COOC performs slightly better than S3NPMI on the TwitterData and HotelReviews datasets. Overall, the results show that sentiment classification benefits much from semantic indexing using $S3$.

Dataset	S3COOC	S3NPMI	S3COOCSWN	S3NPMISWN
MovieReviews	83.4	85.0	85.4^{+}	$\mathbf{85.8}^{+}$
AmazonReviews	78.7	**81.3**	76.8^{-}	81.0
TwitterData	82.9	82.7	84.2^{+}	$\mathbf{85.1}^{+}$
HotelReviews	68.4	67.3	$\mathbf{70.7}^{+}$	68.1^{+}

Table 6.3: Comparison of standard $S3$ and extended $S3$ with sentiment scores from SentiWordNet.

Table 6.3 presents results of of comparing standard $S3$ representations, with $S3$ representations extended with sentiment scores from SentiWordNet. Here also best results for each dataset are shown in bold. Values in the S3COOCSWN and S3NPMISWN columns shown with a $^{+}$ sign represent significant improvement in classification accuracy compared with their non-lexicon based counterparts i.e. S3COOC and S3NPMI respectively while values with $^{-}$ represent a significant depreciation in performance. As expected, the best results are generally achieved using either S3COOCSWN or S3NPMISWN representations. Also, augmenting $S3$ with sentiment scores from SentiWordNet produces significant improvements on all datasets except AmazonReviews where the augmented representation resulted in a statistically significant decline in classification accuracy.

Close examination of AmazonReviews dataset reveals that terms with strong sentiment are often used in documents that belong to the opposite sentiment class. For example, we find the following review in a document belonging to the *Negative* class:

"I just can't imagine how anyone enjoyed this movie"

Note the use of the term 'enjoyed' in the review which has a positive sentiment score of 0.32 in SentiWordNet. Accordingly, adding this score to the representation of documents belonging to the *Positive* class will make short documents such as this one even more similar to the *Positive* documents when provided as a query document. Indeed in our evaluation, S3COOCSWN incorrectly classifies this document as 'Positive' while S3COOC correctly classifies it as *Negative* because S3COOC does not make use of a sentiment lexicon. Thus, in the S3COOC representation, the weights of sentiment terms are a reflection of their distribution in the corpus. A similar problem is also expected in situations where negation is used in a document e.g. "I did not enjoy this movie". This indicates that further contextual analysis is required when working with sentiment lexicons in order to avoid these types of problems. Nonetheless, the significant improvements achieved on most datasets indicate that our approach of augmenting $S3$ with sentiment scores from SentiWordNet is effective for sentiment classification.

Considering the popularity of sentiment analysis on tweets, it is important to discuss the stae-of-the-art in sentiment classification on twitter. Recall from Table 6.2 that our baseline performance on TwitterData is 71.6. This is comparable with the baseline achieved in (Agarwal, Xie, Vovsha, Rambow & Passonneau 2011) (71.35) on a similar binary classification task, using a similar unigram representation with SVM classifier. However, note that the improvements obtained using the $S3$-based representations (82.9 and 82.7 using S3COOC and S3NPMI respectively) and also using the the hybrid approach with SentiWordNet (84.2 and 85.1 using S3COOCSWN and S3NPMISWN respectively) are much higher than the best performance reported in (Agarwal et al. 2011) (75.39). The work in (Go, Bhayani & Huang 2009) presents a higher unigram baseline performance of 82.2 using SVM. In (Lin & Kolcz 2012), a much larger dataset was used (from 1 million to 100 million tweets) which produced baseline unigram accuracies of between 77.5 (for 1 million tweets) to 78.5 for (100 million tweets). However, note that unlike the dataset used in our evaluation and the dataset presented in (Agarwal et al. 2011) where the ground truth sentiment labels were obtained using manual annotation, the ground truths in (Go et al. 2009)

and (Lin & Kolcz 2012) were obtained automatically using the distance supervision technique (Go et al. 2009).

6.5 Chapter Summary

In this chapter, we demonstrated the application of the $S3$ semantic indexing approach to the task of sentiment classification. We also demonstrated how sentiment scores from a sentiment lexicon (SentiWordNet) can be utilised with $S3$ to improve sentiment classification performance. Evaluation shows $S3$ to be very effective for sentiment classification, significantly outperforming baseline BOW representation (without semantic indexing). Furthermore, combining $S3$ with knowledge from a sentiment lexicon significantly improves the performance of $S3$ on sentiment classification.

An important advantage of providing sentiment scores from a lexicon to $S3$ is that sentiment lexicons provide a more general judgement of sentiment strength that is likely to help avoid over fitting the training corpus. Accordingly, we presented an approach that utilises a simple, yet effective linear interpolation of class relevance term weights and class-specific sentiment scores. The use of weighting parameters (α and β) in the combination allows for controlling the contribution from the sentiment lexicon to the final document representation which helps to mitigate against noise from the lexicon.

Chapter 7

Event Extraction for Concept-Based Indexing

All document indexing approaches discussed so far are based on the same underlying assumption, that terms alone are sufficient to model the meaning of text documents. However, for some tasks, a more effective indexing vocabulary is better defined at a higher conceptual level rather than at the lower level of keywords. Unlike keywords, concepts have much more semantic information associated with them. Such semantically rich features are very useful in text classification tasks where the class boundary of a document collection is defined by a semantic distinction rather than topic. That is, the distinction between classes is not based on topic but rather some difference in semantics in the content of the documents. For example in the domain of incident reporting, one may wish to retrieve or categorise incident reports based on say incident cause, whether or not injuries or fatalities are recorded in the report, and whether or not there were damage reported. In such cases, the distinction between document categories is not related to topic, but rather to some specific occurrence or event (i.e. cause, injury, or damage) described in the documents.

To support these types of semantic classification tasks, we turn to event extraction. Events are defined as "a specific occurrence..., something that happens or a change of state"(LDC 2005). These are typically expressed in text using single words (e.g., "fall" and "break"), or multi-word expressions (e.g "take off") (Filatova & Hatzivassiloglou 2003, LDC 2005, Sauri, Goldberg, Verhagen & Pustejovsky 2009). Thus, given an incident report, we can observe that the important conceptual information such as causes, injuries and damages are typically described using event

expressions. Take for example the following snippets extracted from a report about a fire related incident:

"Gas was **leaking** from the pipe"

"This resulted in a **fire**"

"The operator was severely **burned**"

Observe that the expressions '**leaking**', '**fire**' and '**burned**' all satisfy the definition of "specific occurrence" and "something that happened". Thus, these three expressions are considered events and they pretty much tell us the important occurrences in this incident. For example, '**leaking**' tells us the occurrence that caused the incident (a gas leak), '**fire**' tells us what type of incident it was and '**burned**' tells us the consequence of the incident. Accordingly, a proper document index in this situation should not only capture these events, but also assign a high level of importance to them. This is important in order to allow for comparing and classifying incident reports based on incident cause, incident type or damages and injury types reported.

In this chapter, we present an unsupervised heuristic approach for extracting events from the content of documents called RUBEE (RUle-Based Event Extractor). We further present a framework for using events and event polarity (whether the occurrence of the event is negated or affirmed) for text representation with a view to improving text classification performance. Specifically, we study the effectiveness of an event-based representation in differentiating between documents that have very similar context (i.e. describe similar situations) but report different eventualities. For this purpose we present results from an experiment designed to study the categorisation of reports, on the basis of, whether or not injuries were sustained in similar incident scenarios. A comparative study is used to analyse classification performance on document representation with events extracted using RUBEE versus those extracted using a benchmark event extraction system called EVITA (Saurí, Knippen, Verhagen & Pustejovsky 2005).

This chapter is structured as follows: Section 7.1 presents RUBEE, our event extraction algorithm. Our proposed representation framework that uses both semantic and lexical information for document indexing is presented in Section 7.2. Evaluations are presented in Section 7.3. We conclude this chapter with a summary in Section 7.4.

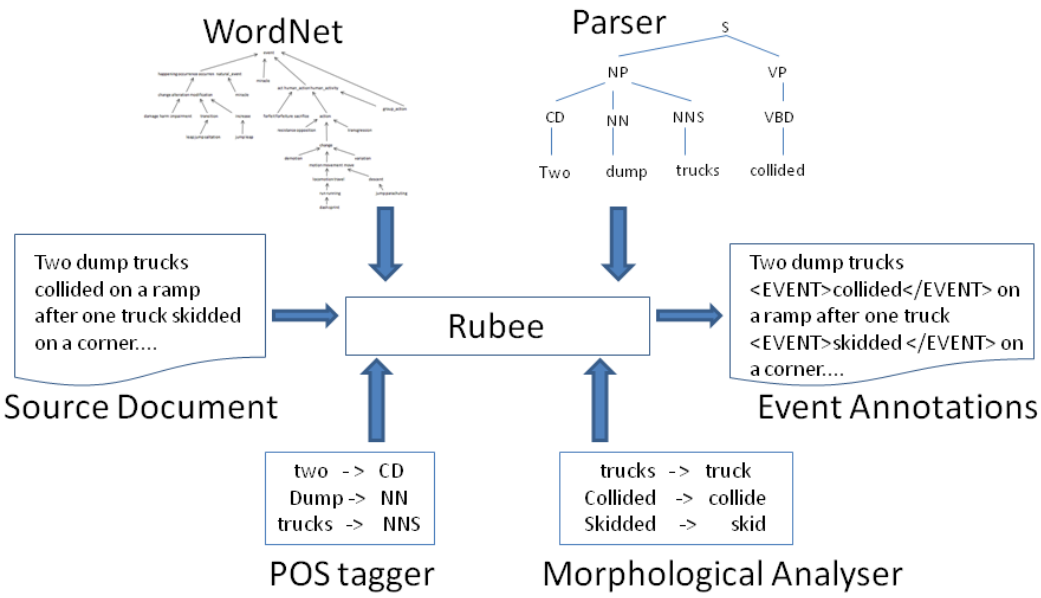

Figure 7.1: Event extraction process.

7.1 RUBEE- RUle-Based Event Extraction

RUBEE is an unsupervised rule-based event extraction algorithm which exploits knowledge from linguistic analysis and a lexical database. A source document is read, tokenized, tagged with part-of-speech information and sentences are parsed into syntactic and dependency structures using the Stanford Parser (Marneffe, Maccartney & Manning 2006). This allows us to identify the grammatical roles of tokens in the sentence e.g., whether a verb is a main verb or an auxiliary. This information is used by RUBEE to decide whether candidate tokens should be accepted or rejected as valid events. The event extraction process is shown in Figure 7.1. Here WordNet (Miller 1995) is used to provide background knowledge for identifying event candidates. For example hypernymy information is used to identify candidate nouns for event extraction. We note that a glossary or ontology of events could also be utilised here instead of Wordnet. However, in the absence of such resources, WordNet provides a satisfactory alternative.

RUBEE's event extraction algorithm appears in Figure 7.2. The function $pos(s)$ returns a sequence of part-of-speech tags for the corresponding tokens in the sequence s. The part of speech tags used in the algorithm (VB, RB, NN etc.) have the same meaning as defined in the Penn Treebank tagset (Santorini 1990). Given a sentence $\mathcal{S}$ a regular expression is matched in order

Let:
$\mathcal{S} = \{t_1, ..., t_n\}$, a sentence which is a sequence of tokens t_i
$s \subseteq \mathcal{S}$, a subsequence of tokens in $\mathcal{S}$
p_i, a part-of-speech (pos) tag for token t_i
$p = \{p_1, ..., p_n\}$, the sequence of all pos tags p_i of tokens t_i in s
$pos(s) \rightarrow p$, a function:
$\simeq$, a regular expression matching operator
$\mathcal{C}$, the set of all candidate event tokens
$\mathcal{E}$, the set of all selected events
$\mathcal{V}$, the set of all verbs in WordNet
$\mathcal{N}$, the set of all identified WordNet noun event Synsets
For each $s \in \mathcal{S}$
  If $pos(s) \simeq$ VB.*RB|VB.*IN|VB.*RP
    If s $\in \mathcal{V}$
      $\mathcal{C} = \mathcal{C} + s$
    Else
      $\mathcal{C} = \mathcal{C} + mainverb(s)$
  Else if $pos(s) \simeq$ VB.*
    $\mathcal{C} = \mathcal{C} + s$
  Else if $pos(s) \simeq$ NN.*
    If $hypernym(s) \in \mathcal{N}$
      $\mathcal{C} = \mathcal{C} + s$
  Else if $pos(s) \simeq$ JJ.*
    If $verbDerived(s)$
      $\mathcal{C} = \mathcal{C} + s$
For each $c \in \mathcal{C}$
  If not $auxilliary(c) \wedge$ not $NN\_modifier(c)$
    $\mathcal{E} = \mathcal{E} + c$
For each $e \in \mathcal{E}$
  $extractPolarity(e)$

Figure 7.2: RUBEE Algorithm

to identify candidate token sequences based on part-of-speech information. Candidate events are then filtered using a sequence of conditional statements to identify the final set of valid events. Finally, the polarity (negative or positive) of each event is identified. We consider event candidates from three parts-of-speech categories: verbs, nouns and adjectives. Corresponding extraction heuristics for each of these part-of-speech categories are explained with examples below followed by a discussion of how polarity information is used for event extraction.

7.1.1 Verbs

Verbs typically express actions or happenings and as such, are good candidates for events. However, we use the following rules to filter out unlikely verb candidates:

- Auxiliaries: Auxiliary verbs are non-main verbs in a clause and typically serve to only support the main verb. For example:

  "Closing the lid would have **prevented** the hot material from falling".

  In the preceding example (and all subsequent examples) the event is shown in bold and the non-event verbs are underlined. The verbs "would" and "have" are auxiliary verbs that modify the main event verb. Thus, only "prevented" is extracted as an event.

- Modifiers: Verbs often appear as modifiers of nouns and noun phrases e.g., "drilling team" and "cutting equipment" Such verbs are not extracted as events.

- Verb+Particle and Verb+Preposition: These types of constructs have a different meaning from their verb component e.g.,

  "The regulator was **turned off** and the fire self extinguished"
  "The fire was **put out** with a hand held extinguisher"

  Such constructs are identified and extracted as events. We validate all extracted verb+particle and verb+prep sequences by looking them up in WordNet. Thereafter, for any sequence of words not known to WordNet (e.g., 'spray over') we extract only the main verb ('spray').

7.1.2 Nouns

Unlike verbs, most nouns are not events. Thus identification of noun events requires a more selective process. A small set of WordNet synsets called event parents, were manually identified and their hyponyms (child nodes) are maintained as relevant event expressions. These synsets were identified by manually extracting noun events from a set of training documents, mapping each one to a corresponding WordNet synset and then identifying a suitable hypernym from the root. A hypernym is suitable if it is considered to denote a type of occurrence or event. For example the noun events "extraction", "combustion" and "absorption" are manually extracted from the

training documents and the synset which subsumes these events is identified in WordNet. In this case this is the synset "Physical Process" which is defined as "a sustained phenomenon or one marked by gradual changes through a series of states". The parent of "Physical Process" is the synset "Physical Entity" which does not fit the description of a type of occurrence or eventuality. Thus we created a rule which accepts nouns that are hyponyms of "Physical Process" as candidate events. The final set of WordNet parent nodes used for selecting nouns are:

- Event: The first sense of event in WordNet is defined as "something that happens at a given time and place". Hyponyms of this synset makes up the largest class of event words e.g., **collision**, **movement** and **fire**.

- Physical Process: This synset is defined as "a sustained phenomenon or one marked by gradual changes through a series of states" and it includes the hyponyms **ignition**, **combustion** and **overheating**.

- Ill Health: This is defined by WordNet as "a state in which you are unable to function normally and without pain". Hyponyms of this synset include the events: **fracture**, **contusion** and **laceration**.

- Symptom (medicine): This has the definition: "Any sensation or change in bodily function that is experienced by a patient". Relevant hyponyms include: **soreness** and **pain**.

- Injury: We ignore the first sense of "injury" because it is already a hyponym of the synset "Ill Health". The second sense of "injury" has the definition"An accident that results in physical damage". Hyponyms of this synset include the event **concussion**. Note that injuries (as well as ill health and symptoms) are extracted as valid events because they fit within the definition of "change of state".

7.1.3 Adjectives

The last class of events types are adjectives, which often occur as participles e.g.,

"A fitter suffered a **lacerated** forehead"

"A light vehicle driver received a **bruised** shoulder"

These event types are extracted with the help of WordNet which is used to identify adjectival

expressions that are derived from verbs. WordNet maintains a “participle of” relation between adjectives and their corresponding root verbs. For example the adjective “elapsed” has a “participle of” relation with the verb “elapse”. However, this strategy was found to have very limited coverage. Instead an alternative strategy was used whereby morphological analysis is used to derive the verb from the adjective before validating with Wordnet. Since participles typically have the same spelling as past-tense verbs, a lemmatiser is used to transform the adjective into a root verb. For example the adjective “fractured” is lemmatised to “fracture”. The lemma is then looked-up in WordNet. If the lemma is a valid verb, the adjective is accepted as a valid event.

7.1.4 Event Polarity

The polarity of an event is negative if the occurrence of the event is explicitly negated in the text and positive otherwise. Negative polarity is often expressed using a negative word e.g.,“not" and “no". Event polarity is particularly important for retrieval because it helps to distinguish between affirmed and negated occurrences of the same event. This helps to avoid false matching of events that have opposite polarity. Take for example the following sentences:

“An operator suffered crush **injuries**"

“No contact with the electricity was made and **no injuries** were sustained"

Without identifying the polarity of injury, the two sentences can incorrectly be considered similar even though the second example clearly negates the occurrence of injuries. Event polarity is extracted using dependency parse information to check for negative modifiers and negations as shown in Figure 7.3. All events that have a negative determiner (“no”), a negation modifier (“not”) or are objects of a word that indicates negation (e.g “avoid”) are considered to have negative polarity. Consequently all events are stored together with their corresponding polarity value which is later utilised in our document representation and comparison strategy.

7.2 Document Indexing using Events

Once events have been extracted, they need to be utilised for document indexing. In this section we present a framework for utilising extracted events for text document representation where documents are represented using both lexical and event features - lexical to capture general context

Let:
$N = \{n_1, ..., n_m\}$, a set of negation words
$\mathcal{E} = \{e_1, ..., e_m\}$, a set of events
For each $e \in \mathcal{E}$
If $hasNegDeterminer(e) \vee hasNegModifier(e)$
$\vee\ isObjectOf(e, n \in N)$
$e = \neg e$

Figure 7.3: Polarity Extraction Algorithm

and events to capture relevant conceptual information. Lexical features are represented using a standard Bag-of-Words (BOW) indexing vocabulary where text is represented in a vector space whose dimensions correspond to individual terms. Similarly, semantic information is represented using a Bag-of-Events (BOE) vector representation where dimensions correspond to the event vocabulary and separate dimensions are used to represent negative and positive polarity instances of the same event. Thus a document is represented as a pair:

$$d = (\vec{t}, \vec{e}) \tag{7.1}$$

Where $\vec{t}$ is the BOW representation and $\vec{e}$ is a BOE representation for the document d. Here any standard text representation scheme such as binary vectors or *tf-idf* vectors can be used for the entries of both $\vec{t}$ and $\vec{e}$. Note that while $\vec{e}$ captures event information, $\vec{t}$ includes important contextual information that may not be captured by $\vec{e}$.

Figure 7.4 illustrates the representation of a sample document using our approach. Note that the positive and negative polarity instances of the term 'injury' are represented using separate dimensions. The weight of each entry in the BOE vector is thus a binary (0,1) or *tf-idf* weight for the respective event. Similarity between documents is thus computed as shown in Equation 7.2.

$$SIM(d_q, d_i) = (1 - \alpha) Sim(\vec{t}_q, \vec{t}_i) + \alpha Sim(\vec{e}_q, \vec{e}_i) \tag{7.2}$$

Where $SIM(d_q, d_i)$ is the global similarity between a query document d_q and any document d_i from the training corpus, $Sim(\vec{t}_q, \vec{t}_i)$ is the BOW similarity between d_q and d_i, $Sim(\vec{e}_q, \vec{e}_i)$ is the BOE similarity between d_q and d_i, and α is a mixing parameter. Thus the similarity between two

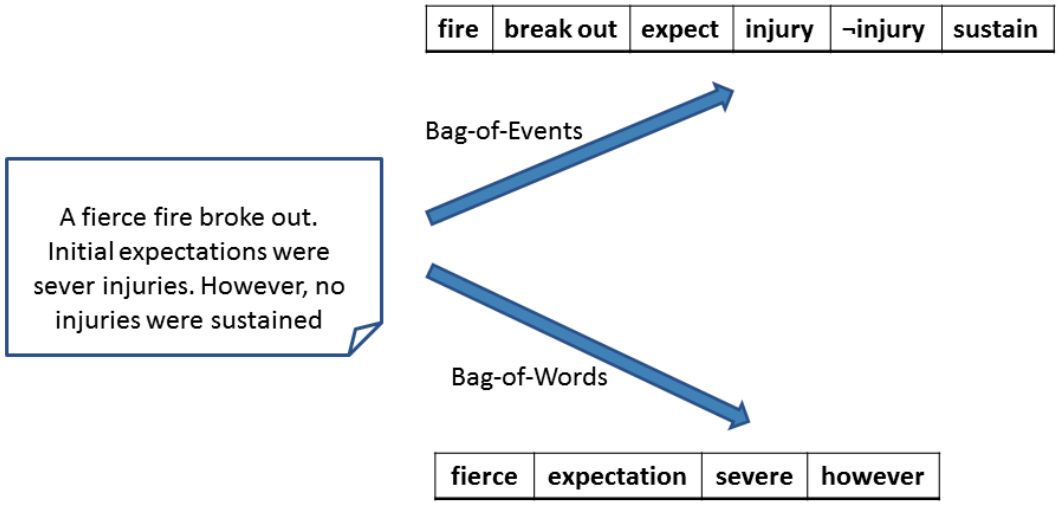

Figure 7.4: Representation of a document using BOW and BOE vectors.

documents is an aggregation of their terms and events similarities, whilst α controls the contribution of each representation's similarity to overall global similarity. Note that increasing the value of α increases the contribution of the BOE representation. Both $Sim(\vec{t}_q, \vec{t}_i)$ and $Sim(\vec{e}_q, \vec{e}_i)$ are obtained using the cosine similarity measure.

7.3 Evaluation

The aim of our experiments is to establish the utility of event-based semantic indexing for classification of incident reports. Our comparative study is applied to the following representation schemes:

1. BOW: a BOW-only representation where $\alpha = 0$
2. BOE: a BOE-only representation where $\alpha = 1$
3. $Comb$: a combined representation where $0 < \alpha < 1$

We also wish to asses how our event extraction algorithm (RUBEE) compares to an alternative event extraction approach, EVITA (Saurí et al. 2005). EVITA is a system for identifying and extracting events from text using a combination of linguistic analysis, heuristic rules and lexical lookup. One of the key differences between EVITA and RUBEE involves the manner in which

sentences are processed. While RUBEE uses full dependency parsing, EVITA uses chunking; a form of shallow parsing that produces linguistically defined groups of adjacent words e.g., noun phrases and verb phrases, rather than full parse trees. Although chunking is a less expensive operation compared to parsing, parse trees provide richer syntactic information of sentences and so are more useful for deep linguistic analysis. Consequently, EVITA's rules are based on pattern matching on word sequences while RUBEE's rules are based on dependency-tree structures. Another key difference between RUBEE and EVITA is that unlike RUBEE, EVITA does not recognise verb+particle and verb+prep event types. When such constructs are encountered, EVITA extracts only the head verb.

Noun events are extracted by EVITA based on hypernymy information from WordNet. A total of twenty five WordNet subtrees are used for this purpose and any noun event candidate corresponds to a synset in any of these synsets is accepted as a valid event. However, details of the synsets used are not given. Also, extraction of adjectival events in EVITA is based on lookup whereby candidate adjectival events are accepted if they occur in the list of annotated events in the TimeBank-1.2 Corpus which contrasts with the use of morphological analysis by RUBEE. Attributes of events including polarity are extracted by EVITA using pattern matching techniques. However, details of these pattern matching techniques are not given.

We also compare with a baseline event extraction technique that extracts all non-stopword verbs as events without further linguistic analysis. Thus, we compare performance of BOE representations generated using the following event extraction approaches:

1. VERBS: a baseline approach that extracts only verbs as events according to part-of-speech information without further linguistic analysis

2. RUBEE (see Section 7.1)

3. EVITA: A benchmark event extraction system presented in (Saurí et al. 2005)

Accordingly, BOE representations obtained using the different event extraction approaches are called $\boldsymbol{BOE_{VERBS}}$, $\boldsymbol{BOE_{RUBEE}}$ $\boldsymbol{BOE_{EVITA}}$ respectively. The representation obtained by combining events from RUBEE and BOW as described in Section 7.2 is called $\boldsymbol{Comb_{RUBEE}}$. Currently we determine the alpha value that results in best value empirically. We report text classification accuracy using a with 3 nearest neighbours averaged over 5 runs of stratified 10-fold

Name	*Domain*	*Description*	*Voc. Size*
TRUCKC	TruckCollision	Incidents involving truck collision	1182
Fire	Fire	Incidents involving fire outbreak	1326
TRUCKR	TruckRollover	Incidents involving truck rollover	1031
LIGHTV	LightVehicle	Incidents involving light vehicle accidents	1064
MISCI	MiscIncidents	Miscellaneous incidents	1581
ROLLCOL	RolloverCollision	A combination of TruckR and TruckC incidents	1212

Table 7.1: Datasets

cross-validation experiments. Significance is reported from a t-test with 95% confidence.

7.3.1 Datasets

Several benchmark datasets were created using incident reports crawled from the Government of Western Australia's Department of Mines and Petroleum website [1] [2]. These incident reports are pre-classified into "Injury" and "NoInjury" classes. Accordingly we treat this as a classification task. Details of these datasets are given in Table 7.1. We also combine the TRUCKR and TRUCKC datasets to form a new dataset called ROLLCOL. This new dataset is used to further test if event information can help distinguish between collision and rollover incidents involving trucks. Each dataset in Table 7.1 contains 200 documents; 100 documents in each class. This includes the ROLLCOL dataset which contains 100 in each class selected at random from the TruckRollover and TruckCollision datasets respectively. All have a similar vocabulary size (with MISCI having the largest vocabulary) from which the indexing vocabulary will be drawn for each algorithm.

7.3.2 Results

From table 7.2, we observe that event-only representation with BOE_{RUBEE} was significantly better than BOW on 4 of the datasets. Performance of BOE_{RUBEE} on the RollCol dataset is not significantly better than BOW while BOW is significantly better than both BOE_{RUBEE} and BOE_{EVITA} on the MiscI dataset. The reason for this might be explained by the variety of different types of incidents and injuries in this dataset introducing a degree of sparseness into the BOE representation. BOE_{RUBEE} significantly outperforms BOE_{EVITA} on all datasets except the RollCol dataset where BOE_{EVITA} performs slightly (but not significantly) better. BOE_{VERBS}'s

[1] http://dmp.wa.gov.au

[2] Available for download at: http://bit.ly/1qtuFUo

	TruckC	*Fire*	*TruckR*	*LightV*	*MiscI*	*RollCol*
BOW	80.5	84.7	78.4	81.0	84.7	83.4
BOE_{VERBS}	78.5	83.4	76.7	75.3	75.6	81.1
BOE_{EVITA}	80.8	82.7	74.4	81.3	78.6	87.1
BOE_{RUBEE}	84.5	**90.0**	85.4	85.1	81.0	85.2
$Comb_{RUBEE}$	**87.5**	**90.0**	**86.4**	**88.1**	**88.6**	**91.1**

Table 7.2: Classification accuracies of different representation schemes. Best results on each dataset are presented in bold.

performance was generally poor compared to all other approaches including BOW. This shows that the linguistic analysis used by the event extraction algorithms is important for correctly identifying event information for document indexing.

For the combined representation ($Comb_{RUBEE}$), we observed improvements over all 4 individual indexing schemes on all 6 datasets. Specifically, $Comb_{RUBEE}$ performed significantly better than BOW, BOE_{VERBS} and BOE_{EVITA} on all datasets. Comparing with BOE_{RUBEE}, $Comb_{RUBEE}$ performed significantly better on all datasets with the exception of FIRE and TRUCKR. This confirms our hypothesis that the lexical information in the BOW representation and the semantic information in the BOE representation are complementary. Thus, a combination of both leads to even better retrieval performance.

To further motivate the need for document indexing using semantically rich concepts rather than using only terms, we include a comparative evaluation of $Comb_{RUBEE}$ with semantic indexing using the RWSI and $S3$ frameworks on term-based (BOW) representations. Accordingly, we compare $Comb_{RUBEE}$ with the following representations:

- RWSI: Semantic indexing on BOW representation using RWSI framework (see Chapter 4)
- $S3$: Supervised Semantic indexing on BOW representation using $S3$ framework (see Chapter 5)

Results of this comparative evaluation are presented in Table 7.3. Observe how the improvements from semantic indexing using both RWSI and $S3$ are much less than that from $Comb_{RUBEE}$. The limited improvements from Semantic indexing is due to the fact that the discriminatory semantics between the two classes is not well captured by co-occurrence statistics. This is expected because, unlike topic classification, the distinction between the two classes in each dataset is due to the presence and absence, as well the affirmation and negation of certain events, rather than a

	TruckC	*Fire*	*TruckR*	*LightV*	*MiscI*	*RollCol*
BOW	80.5	84.7	78.4	81.0	84.7	83.4
RWSI	81.2	83.2	80.8	81.5	82.8	85.7
$S3$	83.0	84.2	82.4	86.8	84.9	87.8
$Comb_{RUBEE}$	**87.5**	**90.0**	**86.4**	**88.1**	**88.6**	**91.1**

Table 7.3: Comparison of $Comb_{RUBEE}$ with term-based semantic indexing. Best results on each dataset are presented in bold.

distribution of terms. The superior results achieved using $Comb_{RUBEE}$ further demonstrates the effectiveness of our event indexing approach for semantic text classification.

In Table 7.4 we present results for RUBEE with and without polarity information. Improvements are realised with polarity information on all datasets except FIRE and MISCI. Improvements on the TRUCKC and TRUCKR datasets are statistically significant. Table 7.5 provides statistics of negations found in each dataset. Observe that in the FIRE datasets, a total of 8 events were found with negative polarity. However, none of these were negations of injury events and thus, no benefit was realised on classification accuracy. In contrast, 25 negations were extracted from the TRUCKR dataset, 14 of which were negations of injuries. This leads to significantly better classification accuracy on the TRUCKR dataset.

	TruckC	*Fire*	*TruckR*	*LightV*	*MiscI*	*RollCol*
BOE_{RUBEE}	**84.5**	**90.0**	**85.4**	**85.1**	81.0	**85.2**
$BOE_{RUBEE}(NoPol)$	82.7	89.9	81.7	84.2	**81.6**	84.8

Table 7.4: Classification accuracy of RUBEE with and without event polarity.

For the ROLLCOL dataset, a total of 46 negations were found in Table 7.5, 15 of which are negations of injuries. However, recall that the task on this particular dataset is to distinguish between "Collision" and "Rollover" incidents. Thus negations of injuries are found in both classes and are not useful for distinguishing between different classes. Also, unlike "Injury" and "NoInjury" classes, "Collision" and "Rollover" incidents are not polar opposites. Consequently, out of all negations found, none were negations of "Collision" or "Rollover" events. This further suggests that polarity information is particularly useful for distinguishing between classes that are polar opposites.

Figure 7.5 shows average accuracy for increasing values of α over all runs of the RUBEE algorithm. Best results are generally obtained within the range $0.4 \leq \alpha \geq 0.7$. This indicates

	TruckC	*Fire*	*TruckR*	*LightV*	*MiscI*	*RollCol*
Total event negations	42	8	25	27	20	46
Negations of injury events	9	0	14	9	3	15

Table 7.5: Statistics of negations extracted from all datasets

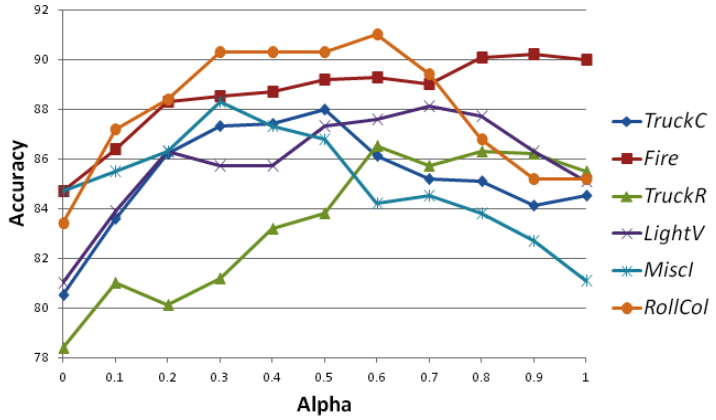

Figure 7.5: RUBEE's performance as a function of α on each dataset

the BOE representation is largely responsible for the improved performance of the Combined approach. The difference between the highest and lowest accuracy obtained between $\alpha = 0.1$, and $\alpha = 0.9$ (i.e excluding BOW-only and BOE-only representations) is from 3.8% for Fire to 6.4% for the TruckR. However, note that (with the exception of the MiscI and RollCol datasets) the variation in accuracy levels-off with higher values of alpha ($\alpha >= 0.5$).

These results demonstrate the utility of event extraction for representing textual documents in domains characterised by eventualities. The results also confirm our proposed document representation model effectively combines contextual information from terms with semantic information from events. Lastly, the comparison between RUBEE and EVITA on these tasks points in favour of RUBEE as an effective event extraction system.

7.3.3 Application of RUBEE to New Domain of Aviation Incidents

To further verify the effectiveness of our events-based indexing approach, and to test its portability, we include experiments on a dataset of aviation incident incident reports crawled from the Skybrary website [3]. Incident reports in this website are tagged with information to categorise the reports according to cause e.g. bird strike and weather. Reports are also tagged with information

[3] http://www.skybrary.aero/index.php/Main_Page

on the whether damages and injuries occurred in the incident. Accordingly we create a binary-class dataset with the classes 'Damages Injury' and 'No Damages Injury'. The dataset contains 200 documents partitioned equally between the two classes. In Table 7.6, we compare the results of a baseline BOW representation, semantic indexing using RWSI and $S3$ and also $Comb_{RUBEE}$.

	Skybrary
BOW	64.0
RWSI	64.5
$S3$	65.5
$Comb_{RUBEE}$	**68.5**

Table 7.6: Comparison of $Comb_{RUBEE}$ with term-based semantic indexing on the Skybrary dataset.

As can be observed, the combined event and terms based index ($Comb_{RUBEE}$) out performs all the other representation approaches on the Skybrary dataset as well. This indicates the utility of our events-based representation approach, regardless of domain. This also indicates the effectiveness of RUBEE on domains other the one it was originally developed on.

7.4 Chapter Summary

In this chapter we have demonstrated the utility of event information for concept-based indexing of incident reports. Indexing of incident reports using events allows for comparing and classifying incident documents based on incident cause, type of injury and type damages reported. Achieving this requires that the indexing vocabulary includes semantic features to capture relevant events and their attributes. Accordingly, we presented an unsupervised heuristic approach for the extraction of atomic events called RUle-Based Event Extractor (RUBEE). RUBEE uses linguistic analysis and a lexical database, WordNet, to identify events and their attributes directly from textual content.

We also presented a general framework for the indexing of text using both lexical and event information. Our framework uses a weighting parameter to control the strength of the contribution from the lexical and event parts of the document representation to the global similarity between documents. We also demonstrated how event polarity (whether or not the occurrence of an event is negated) can be included in the document index to distinguish between asserted and negated occurrences of the same event.

Our evaluation compared text classification performance on document representations pro-

duced using event-only, term-only and combined (events and terms) indexing vocabularies. Results show the events only representation to significantly out-perform a term-only representation, while the combined representation significantly out-performed both term-only and event-only representations. The high accuracy of the combined approach is because, while events are useful for capturing semantic information, terms are useful for capturing additional context. Thus, the combined representation is able to leverage both semantic information and important contextual information for improved classification accuracy. Results also show the inclusion of event polarity to lead to significant improvement in classification performance.

Our evaluation also compares event representation using RUBEE and a benchmark event extraction algorithm, EVITA, as well as a baseline event extraction approach that uses only verbs. Results show event information extracted using RUBEE to out perform the two others in text classification accuracy. Also, our evaluation shows the use of polarity information to significantly improve the performance of the event-based representation.

Chapter 8

Conclusion

In this thesis we addressed the problem of document indexing in the Vector Space Model (VSM) for text classification. We identified three main problems with the standard VSM that limits its performance for text classification. The first problem is the term independence which makes the VSM susceptible to variation in indexing vocabulary. The second problem of the VSM for text classification is the lack of supervision, where class knowledge is ignored in the process of generating document vectors. Thirdly, the standard VSM utilises a term-only indexing vocabulary for document representation. However, for certain tasks, terms are not sufficient to model the semantics needed for accurate document classification. Accordingly, we presented comprehensive analyses that provide insight into the limitations of the current state-of-the-art, and also introduced frameworks and algorithms that address these limitations. This chapter summarises our main contributions and highlights future directions.

8.1 Contributions

In the following subsections, we revisit our objectives and examine the extent to which these have been achieved.

8.1.1 Analysis of the Performance of Semantic Indexing for Text Classification

In chapter 3, we presented a detailed evaluation of semantic indexing with semantic relatedness knowledge extracted using both knowledge-resource-based, and distributional approaches. Four knowledge-resource-based approaches were considered, Wu & Palmer, Lin, Leacock & Chodorow

and Jiang & Conrath. All four approaches use WordNet for computing semantic relatedness between vocabulary terms and the resulting values were used for semantic indexing. The result of text classification on 25 datasets showed very little improvement from using any of the four knowledge-resource-based approaches. Note that while these WordNet based metrics have been widely evaluated on linguistic tasks such as synonymy detection and word pair association, to the best of our knowledge, this is the first time such a comprehensive evaluation has been reported using these metrics on text classification.

For the distributional semantic relatedness approaches, first order document co-occurrence, pointwise mutual information and latent semantic indexing were used. The performance of the distributional semantic relatedness approaches was much better than the knowledge-resource-based approaches. Nonetheless, the performance of the distributional approaches also revealed that semantic indexing does not always improve text classification performance and may sometimes even be harmful. Our results suggest that datasets with documents written in a more professional and consistent style benefit more from semantic indexing. We also observed that datasets with fewer and shorter documents benefited less from semantic indexing.

Considering that extracting semantic relatedness is a computationally expensive process, we set out to determine when and when not to apply semantic relatedness using meta-learning. Accordingly we presented a case-based approach for predicting when to use semantic indexing. Results show that our case-based approach is able to correctly predict the performance of semantic indexing on a range of datasets with over 80% accuracy. Note again that, to the best of our knowledge, this is the first time any attempt has been made to predict when to apply semantic indexing.

An important consideration when building a case-based system is the choice of attributes for case representation. The attributes we used were obtained from several statistical metrics that capture various important characteristics of text datasets. These range from statistics of document frequencies of terms to measures of clustering of document neighbourhood. The high accuracy achieved in predicting when to use semantic indexing indicates that the attributes used for meta-case representation capture characteristics of text datasets that are predictive of the performance of semantic representation. We further used a genetic algorithm to learn the relative importance of our attributes. The high weight assigned to the Nearest Neighbour Similarity attribute indicates the importance of the structure of a dataset is in determining the performance of semantic indexing. Our findings suggest that that better semantic relatedness can be extracted from datasets that have

less variable vocabulary indicated by a higher Nearest Neighbour Similarity.

8.1.2 Propose a new semantic indexing framework

In Chapter 4, we demonstrated the need to capture the relevance of terms during semantic indexing. Our analysis revealed how term relevance is not captured by standard semantic indexing frameworks and how this adversely affects text classification performance. Thus, an important contribution of this work is providing empirical evidence for how the performance of semantic indexing is adversely affected by the inability to capture global term relevance, which is largely responsible for the inconsistent improvements reported.

Based on our findings, we presented the (Relevance Weighted Semantic Indexing) RWSI framework which introduces relevance weighting into semantic indexing. Our evaluation of the RWSI framework using both binary and *tf-idf* document vectors shows RWSI based representations to perform significantly better than both a baseline Bag-Of-Words (BOW) representation with no semantic indexing, as well as semantic indexing using the GVSM and LSI frameworks. The inconsistent improvements realised using both the GVSM and LSI frameworks which do not use term relevance information further supports our hypothesis that term relevance weighting is not only useful, but necessary for effective semantic indexing

A key advantage of the RWSI framework is that it is flexible enough to be used with any semantic relatedness metric and also any supervised term weighting approach, without restrictions. We demonstrated how term relevance weights can be learned directly from the document collection using standard feature selection algorithms. Given that feature selection is a standard pre-processing step of text classification, the weights computed at the feature selection stage can always be supplied to the RWSI framework without the need to compute a separate set of term relevance weights. Furthermore, individual components of the framework e.g. semantic relatedness or term weighting, can be switched on or off, providing much flexibility and control over the document indexing process.

We also demonstrated how the RWSI framework can be used exclusively for supervised document indexing using the Relevance Weighted Indexing (RWI) approach. A comparative evaluation of text classification performance on document vectors produced using unsupervised *tf-idf* indexing versus supervised term weighting using both our RWSI framework and the popular tf-$\delta(t)$ approach was presented. While tf-$\delta(t)$ replaces idf with a supervised weighting component, our

RWI approach combines the supervised weighting with standard *tf-idf*. The superior performance achieved by our approach indicates that, contrary to the tf-$\delta(t)$ assumption, idf and supervised weighting are both important and complementary for document indexing. This also indicates that our RWI approach is able to successfully leverage the best of idf and supervised weighting for improved text classification performance.

8.1.3 Develop a Supervised Semantic indexing Framework

In Chapter 5, we introduced a novel technique called Supervised Sub-Spacing ($S3$) for introducing supervision into semantic indexing. The key idea of $S3$ is to create separate sub-spaces for each class within which semantic indexing transformations are applied exclusively to documents that belong to that class. In this way, $S3$ is able to modify document representations such that documents that belong to the same class are made more similar to one another while, at the same time, reducing their similarity to documents of other classes.

$S3$ requires a different set of semantic relatedness values and term weights to be extracted for each sub-space. Accordingly, we presented the Class Relevance Weighting (CRW) function for learning class-specific term weights. CRW uses Bayesian probabilities to estimate the class relevance of a term as the probability that a document belonging to that class contains the term. We presented a comparative analysis of the CRW function with other alternatives e.g. class-specific probabilities and mutual information. We showed both visually and with the aid of an example, how CRW provides a better model of term relevance compared to the other two alternatives which both tend to under-represent the importance of terms. Furthermore, we presented visualisations of a typical term-document space before and after $S3$ transformation in order to demonstrate the effect of $S3$ on document representations.

We presented a detailed evaluation of the $S3$ approach on 38 datasets from a variety of different domains including news stories, medical abstracts and online reviews. We investigated applying $S3$ with two semantic relatedness metrics: document co-occurrence (DOCCOOC) and Normalised Point-wise Mutual Information (NPMI). Results show $S3$ leads to improvements in the performance of these two metrics on over 80% of the datasets. We also compared two $S3$-based approaches (S3COOC and S3NPMI) with SVM, a supervised version of Latent Semantic Indexing (SPLSI) that uses a technique called Sprinkling, and a supervised LDA (sLDA). Results show that our $S3$-based approaches outperform SVM, SPLSI and sLDA on over 70% of datasets.

Our $S3$ technique has a number of additional advantages compared to the other supervised semantic indexing approaches. Firstly, unlike sLDA and SPLSI, $S3$ is not tied to any specific semantic relatedness approach (i.e. LDA with SLDA, and LSI with SPLSI). We demonstrated this by using $S3$ with both DOCCOOC and NPMI semantic relatedness approaches. Secondly, unlike sprinkling, $S3$ does not require higher order term relations. This means $S3$ does not apply restrictions to the type of semantic relatedness metric that can be used. A third advantage is that $S3$ does not require any parameter tuning whereas sprinkling requires a predetermined number k of artificial terms to be injected into the vocabulary while sLDA requires the optimum number of topics to be determined. In both cases, it is unlikely that globally optimum parameter settings exists and thus, the optimum number of sprinkled terms as well as the optimum number of topics will have to be determined individually for each dataset which further contributes to the complexity of these approaches. Finally, $S3$ requires less computer memory to execute as the term-document space of each individual class gets processed separately which also makes it convenient for distributed and parallel processing. Thus, $S3$ is better suited for real-word commercial applications where the processing cost of LSI and LDA has been a barrier to adoption.

8.1.4 Investigate the Application of our Semantic Indexing Frameworks to Sentiment Classification

In Chapter 6, we presented a case study of applying our $S3$ approach to the task of sentiment classification. $S3$ is able to produce document representations that are more effective for sentiment classification by learning semantic relatedness and term weights exclusively from the set of documents belonging to the same sentiment class. Doing so allows $S3$ to emphasise the semantic associations of terms belonging to same sentiment category in document representations.

Sentiment lexicons have proved very useful for providing the sentiment scores of terms with respect each sentiment category. Thus, sentiment lexicons provide an opportunity to utilise sentiment scores for semantic indexing. Accordingly, we presented an extension of the $S3$ framework for sentiment classification that utilises scores from a sentiment lexicon (SentiWordNet) as further evidence for the relevance of sentiment terms to a sentiment class, by combining these scores with the class relevance weights extracted from the corpus. Results from our evaluation show that semantic indexing using $S3$ leads to statistically significant improvement in sentiment classification performance compared to a baseline Bag Of Words (BOW) representation. Furthermore, aug-

menting $S3$ with sentiment scores from SentiWordNet produces significant improvements in text classification performance compared to standard $S3$.

An important advantage of providing sentiment scores from a lexicon to $S3$ is that sentiment lexicons provide a more general judgement of sentiment strength that is likely to help avoid over fitting the training corpus. Our approach uses a simple, yet effective linear interpolation of class relevance term weights and class-specific sentiment scores. This provides flexibility for controlling the contribution from the sentiment lexicon to the final document representation using a weighting parameter which is useful for mitigating against noise from the lexicon.

8.1.5 Explore the Use of Semantic Concepts e.g. Events for Document Indexing

In Chapter 7 we demonstrated the utility of event information for semantic indexing. Indexing of incident reports using event information allows for comparing incidents based on incident cause, the type of incident or the type of injury. To address this requirement it is necessary to ensure that the indexing vocabulary includes semantic features to capture relevant events and their attributes. Accordingly, we presented an unsupervised heuristic approach for the extraction of events called RUle-Based Event Extractor (RUBEE). RUBEE uses natural language processing together with a set of rules for extracting events and their attributes from the content of a given text document.

We also presented a general framework for the indexing of documents using both lexical and event information. This framework represents a document using two vectors, a regular Bag-Of-Words (BOW) vector consisting of terms and a Bag-Of-Events (BOE) vector comprised of the events extracted from the document. Thus, the similarity between two documents is a combination of their BOW and BOE similarities where weighting parameters are used to control the strength of the contribution from the lexical and event parts of the representation. We further demonstrated how event polarity (whether or not the occurrence of an event is negated) is included in the BOE vector index to distinguish between asserted and negated occurrences of the same events.

We demonstrated the effectiveness of using events for document indexing by comparing text classification performance on document vectors produced using event-only, term-only and combined (events and terms) indexing vocabularies. Results showed that BOE representations significantly out-perform BOW representations, while the combined (BOW and BOE) representation significantly out-performed both BOW and BOE representations individually. The high accuracy of the combined approach indicates that while events are useful for capturing semantic informa-

tion, terms are also useful for capturing additional context. Results also show the inclusion of event polarity to lead to significant improvement in classification performance.

We also demonstrated the utility of events extracted by our RUBEE algorithm by comparing classification performance of BOE document representations indexed with events extracted using RUBEE; a benchmark event extraction algorithm called EVITA; and a baseline event extraction approach that uses only verbs. Results show documents indexed with events extracted using RUBEE to out perform the other two event extraction approaches. We demonstrated the portability of both RUBEE and and our events representation approach by applying both to a dataset of aviation incident reports. The superior performance from our events-based representation further supports the utility of event information for document indexing. This also supports the effectiveness of RUBEE for event extraction on domains other than one the algorithm was developed on.

8.2 Future Work

In this section we highlight some of the limitations of the work we presented in this thesis and also point out some desirable future extensions. Firstly, the case-based approach we presented in Chapter 3 for predicting when to use semantic indexing requires a case base of datasets where the performance of semantic indexing is known on each dataset. However, acquiring such a case base is non-trivial. Doing this requires an adequate number of text classification datasets to be collected and semantic indexing applied on each one which can be quite an expensive undertaking. In the future, it would be desirable to investigate less expensive alternatives for predicting the performance of semantic indexing. Our analysis has already provided insight into the importance of the structure of the neighbourhood of datasets as an attribute to our case-based system. Thus, further study may reveal insights into additional attributes of datasets that can be used to further improve the prediction of the performance of semantic indexing.

Social media data e.g tweets, present interesting opportunities for text classification. Unlike conventional documents, tweets have a high usage of emoticons, metadata tags and URLs. Tweets are also characterised by high usage of abbreviations and slang. Rather than being regarded as noise, these unconventional tokens typically contain rich semantics and valuable information. For example, for sentiment analysis, sentiment labels of tweets have been learned automatically in an unsupervised fashion from emoticons (Marchetti-Bowick & Chambers 2012). Thus, these types

of tokens present an important opportunity for learning additional semantic information that can be further utilised for semantic indexing. Therefore, an important research question is how can semantic information be learned from emoticons, hashtags and other unconventional tokens in tweets, order to improve classification of these types of data.

Contextual analysis is very important for accurate sentiment classification and is now a standard component of many lexicon-based sentiment classification approaches. In deed, our application of the supervised sub-spacing ($S3$) framework to sentiment classification revealed how the presence of positive terms in negative documents and vice-versa can have an adverse effect on sentiment classification accuracy. Recent approaches in sentiment classification have proposed taking into account contextual valence shifters for improved sentiment classification accuracy (Kennedy & Inkpen 2006). Three types of valence shifters are usually considered: negations, which reverse the sentiment polarity of a term; and intensifiers and diminishers which increase and decrease respectively, the degree of sentiment associated with a term. Thus, an important extension of our work would be to investigate how such contextual information can best be included in the representations of documents for use with approaches such as $S3$.

Events have so far proved useful for document indexing, particularly, for classification tasks where the class boundary is based on semantic criteria rather than topic. Also, taking into account event attributes such as negation has shown further improvement in classification accuracy. Thus, this provides a promising direction for future research into what other semantic concepts can be included and what other attributes need to taken into account to support even more sophisticated classification tasks. For example, in the incident reports domain, one may want to classify documents according to the number of people injured in this incident. This would require being able to identify the concept "victim" and also being able to identify the relationship between "victim" and "injury". Indeed this it the ultimate aim of information extraction, to enable all entities, events and relationships to be identified in documents, and to be able to use this information to support sophisticated reasoning. Thus, our work on event extraction provides a useful baseline for the potential of information extraction in supporting more sophisticated text classification tasks.

An important future consideration is the applicability of the semantic indexing frameworks developed in this thesis to multimedia data types. Indeed, LSA has shown great promise in hybrid representations of music using tags and content analysis (Horsburgh, Craw & Massie 2012). This is an example of an interesting trend in multimedia representation where concepts from text rep-

resentation are increasingly being adopted with much success. Another example is the use of the Bag-of-Visual-Words representation for videos (Wang, Song & Elyan 2012) and images (Kaliciak, Song, Wiratunga & Pan 2012), based on the Bag-Of-Words model for text. Thus, given the success of the frameworks developed in this thesis on textual data, it would be interesting to investigate the application of these frameworks to multimedia data.

Bibliography

Agarwal, A., Xie, B., Vovsha, I., Rambow, O. & Passonneau, R. (2011). Sentiment analysis of twitter data, *Proceedings of the Workshop on Languages in Social Media*, LSM '11, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 30–38.

Aggarwal, C. C. & Zhai, C. (eds) (2012). *Mining Text Data*, Springer.

Baccianella, A. E. S. & Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining, *Proceedings of the 7th conference on International Language Resources and Evaluation (LREC'10)*, European Language Resources Association (ELRA), Valletta, Malta.

Bai, B., Weston, J., Grangier, D., Collobert, R., Sadamasa, K., Qi, Y., Chapelle, O. & Weinberger, K. (2009). Supervised semantic indexing, *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, ACM, New York, NY, USA, pp. 187–196.

Banerjee, S. & Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness, *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 805–810.

Basili, R., Cammisa, M. & Moschitti, R. (2005). A semantic kernel to classify texts with very few training examples, *Proceedings of the Workshop on Learning in Web Search, at the 22nd International Conference on Machine Learning (ICML 2005)*.

Bensusan, H., Giraud-Carrier, C. & Kennedy, C. (2000). A higher-order approach to meta-learning, *Proceedings of the ECML'2000 workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, pp. 109–117.

Blei, D. & McAuliffe, J. (2008). Supervised topic models, *in* J. Platt, D. Koller, Y. Singer & S. Roweis (eds), *Advances in Neural Information Processing Systems 20*, MIT Press, Cambridge, MA, pp. 121–128.

Blitzer, J., Dredze, M. & Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification, *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Association for Computational Linguistics, Prague, Czech Republic, pp. 440–447.

Bollegala, D., Matsuo, Y. & Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines, *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*, ACM, pp. 757–766.

Boyd-graber, J., Fellbaum, C., Osherson, D. & Schapire, R. (2006). Adding dense, weighted connections to wordnet, *Proceedings of the 3rd International WordNet Conference*.

Budanitsky, A. & Hirst, G. (2006). Evaluating wordnet-based measures of lexical semantic relatedness, *Computational Linguistics* **32**(1): 13–47.

Cachopo, A. C. (2007). *Improving Methods for Single-label Text Categorization*, PhD thesis, Universidade Tecnica De Lisboa Instituto Superior Tecnico.

Cardoso-cachopo, A., Tulisbon, I., Av, I. & Pais, R. (2007). Combining lsi with other classifiers to improve accuracy of single-label text categorization, *Proceedings of the 1st European Workshop on Latent Semantic Analysis in Technology Enhanced Learning (EWLSATEL 200)*.

Caropreso, M. F., Matwin, S. & Sebastiani, F. (2000). Statistical phrases in automated text categorization, *Centre National de la Recherche Scientifique, Paris, France* .

Castells, P., Fernandez, M. & Vallet, D. (2007). An adaptation of the vector-space model for ontology-based information retrieval, *IEEE Transactions on Knowledge and Data Engineering* **19**: 261–272.

Chakraborti, S., Lothian, R., Wiratunga, N. & Watt, S. (2006). Sprinkling: Supervised latent semantic indexing, *in* M. Lalmas, A. MacFarlane, S. RÃijger, A. Tombros, T. Tsikrika & A. Yavlinsky (eds), *Advances in Information Retrieval*, Vol. 3936 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 510–514.

Chakraborti, S., Mukras, R., Lothian, R., Wiratunga, N., Watt, S. & Harper, D. (2007). Supervised latent semantic indexing using adaptive sprinkling, *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pp. 1582–1587.

Chakraborti, S., Wiratunga, N., Lothian, R. & Watt, S. (2007). Acquiring word similarities with higher order association mining, *Proceedings of the 7th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development*, Springer, pp. 61–76.

Chen, C.-Y., Yeh, J.-Y. & Ke, H.-R. (2010). Plagiarism detection using rouge and wordnet, *Computing Research Repository (CoRR)* **abs/1003.4065**.

Church, K. W. & Hanks, P. (1990). Word association norms, mutual information, and lexicography, *Computational Linguistics* **16**(1): 22–29.

Cilibrasi, R. L. & Vitanyi, P. M. B. (2007). The google similarity distance, *IEEE Transactions on Knowledge and Data Engineering* **19**: 370–383.

Cohen, W. W. (1995). Fast effective rule induction, *Proceedings of the 12th International Conference on Machine Learning (ICML'95)*, pp. 115–123.

Colas, F. & Brazdil, P. (2006). Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks, *in* M. Bramer (ed.), *Artificial Intelligence in Theory and Practice*, Vol. 217 of *IFIP International Federation for Information Processing*, Springer US, pp. 169–178.

Collins, A. M. & Loftus, E. F. (1975). A spreading-activation theory of semantic processing., *Psychological review* **82**(6): 407.

Cristianini, N., Shawe-Taylor, J. & Lodhi, H. (2002). Latent semantic kernels, *Journal of Intelligent Information Systems* **18**(2-3): 127–152.

Cummins, L. & Bridge, D. (2011). On dataset complexity for case base maintenance, *Proceedings of ICCBR*, Springer, pp. 47–61.

Dang, Y., Zhang, Y. & Chen, H. (2010). A lexicon-enhanced method for sentiment classification: An experiment on online product reviews, *IEEE Intelligent Systems* **25**(4): 46–53.

Debole, F. & Sebastiani, F. (2003). Supervised term weighting for automated text categorization, *Proceedings of the 2003 ACM Symposium on Applied Computing*, SAC '03, ACM, New York, NY, USA.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W. & Harshman, R. A. (1990). Indexing by latent semantic analysis, *Journal of the American Society of Information Science* **41**(6): 391–407.

Deng, Z.-H., Tang, S.-W., Yang, D.-Q., Li, M.-Y. & Xie, K.-Q. (2004). A comparative study on feature weight in text categorization, *Advanced Web Technologies and Applications*, Vol. 3007 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 588–597.

Duan, K. & Keerthi, S. S. (2005). Which is the best multiclass svm method? an empirical study, *Proceedings of the Sixth International Workshop on Multiple Classifier Systems*, pp. 278–285.

Fernandez, M., Cantador, I., Lopez, V., Vallet, D., Castells, P. & Motta, E. (2011). Semantically enhanced information retrieval: An ontology-based approach, *Web Semantics: Science, Services and Agents on the World Wide Web* **9**(4): 434 – 452. JWS special issue on Semantic Search.

Filatova, E. & Hatzivassiloglou, V. (2003). Domain -independent detection, extraction, and labeling of atomic events, *In Proceedings of the RANLP Conference*.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification, *Journal of Machine Learning Research* **3**: 1289–1305.

Gabrilovich, E. & Markovitch, S. (2009). Wikipedia-based semantic interpretation for natural language processing, *Journal of Artificial Intelligence Research* **34**: 443–498.

Girill, T. (1985). Online access aids for documentation: a bibliographic outline, *ACM SIGIR Forum*, Vol. 18, ACM, pp. 24–27.

Go, A., Bhayani, R. & Huang, L. (2009). Twitter sentiment classification using distant supervision, *CS224N Project Report, Stanford* pp. 1–12.

Gomez, J. M., Cortizo, J. C., Puertas, E. & Ruiz, M. (2004). Concept indexing for automated text categorization, *in* F. Meziane & E. MÃl'tais (eds), *Natural Language Processing and Information Systems*, Vol. 3136 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 195–206.

Gonzalo, J., Verdejo, F., Chugur, I. & Cigarrin, J. (1998). Indexing with wordnet synsets can improve text retrieval, *COLING/ACM Workshop on Usage of WordNet in Natural Language Processing Systems*, pp. 38–44.

Gottron, T., Anderka, M. & Stein, B. (2011). Insights into explicit semantic analysis, *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, ACM, New York, NY, USA, pp. 1961–1964.

Gracia, J. & Mena, E. (2008). Web-based measure of semantic relatedness, *Proceedings of the 9th International Conference on Web Information Systems Engineering*, WISE '08, Springer-Verlag, Berlin, Heidelberg, pp. 136–150.

Horsburgh, B., Craw, S. & Massie, S. (2012). Music-inspired texture representation, *Proceedings of AAAI*.

Jiang, J. & Conrath, D. (1997). Semantic similarity based on corpus statistics and lexical taxonomy, *Proceedings of the International Conference on Research in Computational Linguistics*, pp. 19–33.

Joachims, T. (1998). Text categorization with support vector machines learning with many relevant features, *European Conference on Machine Learning (ECML '98)*, pp. 137–142.

Kaliciak, L., Song, D., Wiratunga, N. & Pan, J. (2012). Improving content-based image retrieval by identifying least and most correlated visual words, *in* Y. Hou, J.-Y. Nie, L. Sun, B. Wang & P. Zhang (eds), *Information Retrieval Technology*, Vol. 7675 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 316–325.

Kennedy, A. & Inkpen, D. (2006). Sentiment classification of movie reviews using contextual valence shifters, *Computational Intelligence* **22**: 2006.

Kim, H., Howland, P. & Park, H. (2005). Dimension reduction in text classification with support vector machines, *Journal of Machine Learning Ressearch* **6**: 37–53.

Kiryakov, A., Popov, B., Terziev, I., Manov, D. & Ognyanoff, D. (2004). Semantic annotation, indexing, and retrieval, *Journal of Web Semantics* **2**: 49–79.

Lan, M., Tan, C.-L. & Low, H.-B. (2006). Proposing a new term weighting scheme for text categorization, *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, AAAI Press, pp. 763–768.

LDC (2005). *ACE (Automatic Content Extraction) English Annotation Guidelines for Events*, 5.4.3 2005.07.01 edn.

Leacock, C. & Chodorow, M. (1998). *Combining local context and WordNet similarity for word sense identification*, In C. Fellbaum (Ed.), MIT Press, pp. 265–283.

Lewis, D. D., Yang, Y., Rose, T. G. & Li, F. (2004). Rcv1: A new benchmark collection for text categorization research, *Journal of Machine Learning Research* **5**: 361–397.

Lin, D. (1998). An information-theoretic definition of similarity, *Proceedings of the 15th International Conference on Machine Learning*, pp. 296–304.

Lin, J. & Kolcz, A. (2012). Large-scale machine learning at twitter, *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, ACM, pp. 793–804.

Lindner, G. & Studer, R. (1999). Ast: Support for algorithm selection with a cbr approach, *Principles of Data Mining and Knowledge Discovery*, Springer, pp. 418–423.

Liu, B. (2010). Sentiment analysis and subjectivity, *Handbook of Natural Language Processing, Second Edition. Taylor and Francis Group, Boca*.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*, Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers.

Liu, T., Chen, Z., Zhang, B., ying Ma, W. & Wu, G. (2004). Improving text classification using local latent semantic indexing, *In ICDM '04*, pp. 162–169.

Lund, K. & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence, *Behavior Research Methods, Instrumentation, and Computers* **28**: 203–208.

Marchetti-Bowick, M. & Chambers, N. (2012). Learning for microblogs with distant supervision: Political forecasting with twitter, *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 603–612.

Marneffe, M.-C. D., Maccartney, B. & Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses, *Proceedings of International Confrerence on Language Resources and Evaluation*.

Melville, P., Gryc, W. & Lawrence, R. D. (2009). Sentiment analysis of blogs by combining lexical knowledge with text classification, *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, ACM, New York, NY, USA, pp. 1275–1284.

Miller, G. A. (1995). Wordnet: A lexical database for english, *Communications of the ACM* **38**: 39–41.

Mudinas, A., Zhang, D. & Levene, M. (2012). Combining lexicon and learning based approaches for concept-level sentiment analysis, *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, WISDOM '12, ACM, New York, NY, USA, pp. 5:1–5:8.

Muhammad, A., Wiratunga, N., Lothian, R. & Glassey, R. (2013). Contextual sentiment analyis in social media using high-coverage lexicon, *Proceedings of SGAI International Conference on Artificial Intelligence, BCS SGAI*.

Nasir, J. A., Karim, A., Tsatsaronis, G. & Varlamis, I. (2011). A knowledge-based semantic kernel for text classification, *Proceedings of the 18th International Conference on String Processing and Information Retrieval*, SPIRE'11, Springer-Verlag, Berlin, Heidelberg, pp. 261–266.

Ohana, B., Delany, S. & Tierney, B. (2012). A case-based approach to cross domain sentiment classification, *Proceedings of ICCBR*, pp. 284–296.

Padó, S. & Lapata, M. (2007). Dependency-based construction of semantic space models, *Computational Linguistics* **33**(2): 161–199.

Palmer, M., Gildea, D. & Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles, *Comput. Linguist.* **31**(1): 71–106.

Pang, B., Lee, L. & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques, *Proceedings of the ACL-02 Conference on Empirical methods in Natural Language Processing*, EMNLP '02, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 79–86.

Patwardhan, S., Banerjee, S. & Pedersen, T. (2003). Using measures of semantic relatedness for word sense disambiguation, *Computational Linguistics and Intelligent Text Processing*, Vol. 2588 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 241–257.

Peng, Y., Flach, P. A., Soares, C. & Brazdil, P. (2002). Improved dataset characterisation for meta-learning, *Discovery Science*, Springer, pp. 141–152.

Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D. & Goranov, M. (2004). Kim semantic annotation platform, *Journal of Natural Language Engineering* **10**(3-4): 375–392.

Quillan, M. R. (1966). Semantic memory, *Technical report*, DTIC Document.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy, *Proceedings of the 14th International Joint Conference on Artificial intelligence*, pp. 448–453.

Rosso, P., Molina, A., Pla, F., Jimenez, D. & Vidal, V. (2004). Information retrieval and text categorization with semantic indexing, *in* A. Gelbukh (ed.), *Computational Linguistics and Intelligent Text Processing*, Vol. 2945 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 596–600.

Salton, G. & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval, *Information Processing and Management* **24**: 513–523.

Salton, G., Wong, A. & Yang, C. S. (1975). A vector space model for automatic indexing, *Communications of the ACM* **18**: 613–620.

Santorini, B. (1990). Part-Of-Speech tagging guidelines for the Penn Treebank project (3rd revision, 2nd printing), *Technical report*, Department of Linguistics, University of Pennsylvania, Philadelphia, PA, USA.

Sauri, R., Goldberg, L., Verhagen, M. & Pustejovsky, J. (2009). *Annotating Events in English TimeML Annotation Guidelines*.

Saurí, R., Knippen, R., Verhagen, M. & Pustejovsky, J. (2005). Evita: a robust event recognizer for qa systems, *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, Association for Computational Linguistics, pp. 700–707.

Schütze, H., Hull, D. A. & Pedersen, J. O. (1995). A comparison of classifiers and document representations for the routing problem, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, ACM, New York, NY, USA, pp. 229–237.

Scott, S. (1998). Feature engineering for a symbolic approach to text classification, *Technical report*, University of Ottawa.

Scott, S. & Matwin, S. (1998). Text classification using wordnet hypernyms, *Workshop on usage of WordNet in NLP Systems (COLING-ACL '98)*, pp. 45–51.

Sebastiani, F. (2002). Machine learning in automated text categorization, *ACM Computing Surveys* **34**(1): 1–47.

Seco, N., Veale, T. & Hayes, J. (2004). An intrinsic information content metric for semantic similarity in WordNet, Vol. 4.

Smeaton, A. F. (1997). Using nlp or nlp resources for information retrieval tasks, *Natural Language Information Retrieval*, Kluwer Academic Publishers, pp. 99–111.

Steyvers, M. & Griffiths, T. (2007). *Probabilistic Topic Models*, Lawrence Erlbaum Associates.

Strube, M. & Ponzetto, S. P. (2006). Wikirelate! computing semantic relatedness using wikipedia, *Proceedings of the 21st national conference on Artificial Intelligence - Volume 2*, AAAI Press, pp. 1419–1424.

Sun, J.-T., Chen, Z., Zeng, H.-J., Lu, Y.-C., Shi, C.-Y. & Ma, W.-Y. (2004). Supervised latent semantic indexing for document categorization, *IEEE International Conference on Data Mining* **0**: 535–538.

Tsatsaronis, G. & Panagiotopoulou, V. (2009). A generalized vector space model for text retrieval based on semantic relatedness, *Proceedings of the Student Research Workshop at EACL 2009*, pp. 70–78.

Turney, P. D. (2002). Mining the web for synonyms: Pmi-ir versus lsa on toefl, *Computing Research Repository (CoRR)* **cs.LG/0212033**.

Vallet, D., FernÃąndez, M. & Castells, P. (2005). An ontology-based information retrieval model, *in* A. Gomez-Perez & J. Euzenat (eds), *The Semantic Web: Research and Applications*, Vol. 3532 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 455–470.

Vilalta, R., Giraud-Carrier, C., Brazdil, P. & Soares, C. (2004). Using Meta-Learning to Support Data Mining, *International Journal of Computer Science and Applications* **1**(1): 31–45.

Wang, H., Lu, Y. & Zhai, C. (2010). Latent aspect rating analysis on review text data: a rating regression approach, *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, ACM, New York, NY, USA, pp. 783–792.

Wang, L., Song, D. & Elyan, E. (2012). Improving bag-of-visual-words model with spatial-temporal correlation for video retrieval, *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, ACM, New York, NY, USA, pp. 1303–1312.

Weale, T., Brew, C. & Fosler-Lussier, E. (2009). Using the wiktionary graph structure for synonym detection, *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, People's Web '09, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 28–31.

Wei, X. & Croft, W. B. (2006). Lda-based document models for ad-hoc retrieval, *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, ACM, New York, NY, USA, pp. 178–185.

Wettschereck, D., Aha, D. W. & Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms, *Artificial Intelligence Review* **11**(1-5): 273–314.

Wong, S. K., Ziarko, W., Raghavan, V. V. & Wong, P. C. (1987). On modeling of information retrieval concepts in vector spaces, *ACM Trans. Database Syst.* **12**(2): 299–321.

Wu, Z. & Palmer, M. (1994a). Verb semantics and lexical selection, *Proc. of the 32nd annual meeting on Association for Computational Linguistics*, pp. 133–138.

Wu, Z. & Palmer, M. (1994b). Verbs semantics and lexical selection, *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 133–138.

Xu, Z. E., Chen, M., Weinberger, K. Q. & Sha, F. (2012). An alternative text representation to tf-idf and bag-of-words, *Proceedings of the 21st ACM Conferece of Information and Knowledge Management (CIKM)*.

Xue, X.-B. & Zhou, Z.-H. (2006). Distributional features for text categorization, *Proceedings of the 17th European Conference on Machine Learning*, ECML'06, Springer-Verlag, Berlin, Heidelberg, pp. 497–508.

Yan, T., Maxwell, T., Song, D., Hou, Y. & Zhang, P. (2010). Event-based hyperspace analogue to language for query expansion, *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 120–125.

Yang, Y. & Liu, X. (1999). A re-examination of text categorization methods, *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, ACM, New York, NY, USA, pp. 42–49.

Yang, Y. & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization, *Proceedings of the 14th International Conference on Machine Learning*, ICML '97, pp. 412–420.

Zelikovitz, S. & Hirsh, H. (2001). Using lsi for text classification in the presence of background text, *Proceedings Of the 10th ACM International Conference On Information and Knowledge Management*, ACM Press, pp. 113–118.

Zhang, W., Yoshida, T. & Tang, X. (2008). Tfidf, lsi and multi-word in information retrieval and text categorization, *IEEE International Conference on Systems, Man and Cybernetics (SMC '08)*, pp. 108–113.

Appendix A

Publications

- Sani, S., Wiratunga, N., Massie, S., Lothian, R.: Sentiment Classification Using Supervised Sub-Spacing. In: Proc. of SGAI International Conference on Artificial Intelligence, BCS SGAI (2013)
- Sani, S., Wiratunga, N., Massie, S., Lothian, R.: Should Term-Relatedness be Used in Text Representation. In: Proc. of the 22nd International Conference on Case-Based Reasoning, ICCBR (2013)
- Sani, S., Wiratunga, N., Massie, S., Lothian, R.: When to Generalise - A Case-Based Approach to Text Modelling. In: Proc. of the 17th UK Workshop on Case- Based Reasoning, UKCBR (2012)
- Padmanabhan, D., Visweswariah, K., Wiratunga, N., Sani, S.: Two-part segmentation of text documents. In: Proc. of the 21st International Conference on Information and Knowledge Management, CIKM (2012)
- Sani, S., Wiratunga, N., Massie, S., Lothian, R.: Event extraction for reasoning with text. In: Proc. of the 20th International Conference on Case-Based Reasoning, ICCBR (2012)
- Sani, S.,Wiratunga, N., Massie, S., Lothian, R.: Term similarity and weighting framework for text representation. In: Proc. of the 19th International Conference on Case-Based Reasoning, ICCBR (2011)

Appendix B

Experiments with Different Values of k

In this section, we present text classification experiments with similarity weighted kNN using different sizes of k (3, 5, 10, 15 and 20). Accordingly, we compare the following algorithms:

- 3NN - kNN with $k = 3$
- 5NN - kNN with $k = 5$
- 10NN - kNN with $k = 10$
- 15NN - kNN with $k = 15$
- 20NN - kNN with $k = 20$

We apply stop-words removal and lemmatisation text pre-processing operations. Terms with document frequency of less than three are also discarded. Finally, χ^2 feature selection is used to retain only the top 300 terms per dataset for indexing. Cosine function is used for computing similarity.

Results are presented in Table B.1. We report classification accuracy over 5 runs of 10-fold cross validation. Best results in each row are presented in bold font.

	3NN	**5NN**	**10NN**	**15NN**	**20NN**
BactV	85.10	84.21	84.93	**85.89**	85.74
CardR	89.98	89.86	**91.21**	90.82	91.08
NervI	**91.41**	89.89	90.19	89.51	89.58
MouthJ	**89.86**	87.27	88.98	88.31	89.00
NeopE	91.62	91.18	**92.19**	91.63	91.99
DigNut	87.77	88.12	**88.76**	87.87	88.23
MuscleS	83.13	83.40	84.55	**85.15**	84.57
EndoH	91.36	90.48	91.23	91.54	**92.02**
MaleF	**92.33**	91.57	91.23	90.87	90.73
ImmunoV	78.68	78.64	79.34	79.37	**80.06**
NervM	**84.48**	81.46	82.83	83.45	84.03
RespENT	87.23	86.70	88.22	87.93	**88.47**
Hardw	89.81	90.89	90.73	90.92	90.90
MedSp	95.87	**97.51**	97.29	97.12	97.03
CryptE	95.75	**96.94**	96.70	96.05	95.72
ChrisM	88.88	**89.85**	89.50	88.63	88.18
MeastM	94.86	**97.02**	96.99	96.87	96.71
GunsM	93.30	94.94	**95.01**	94.39	93.64
AutoC	94.21	**95.84**	95.40	95.67	95.50
StratM	88.56	89.45	89.98	90.15	**90.25**
EntTour	**94.84**	94.50	94.73	94.41	94.19
EqtyB	**95.77**	94.71	94.62	95.04	95.20
FundA	**90.33**	89.18	89.73	90.31	90.11
InRelD	92.58	92.29	**92.85**	92.52	92.09
NProdRes	85.84	86.27	**87.10**	86.44	86.41
ProdNP	87.76	88.08	**88.16**	87.96	87.54
OilGas	87.19	**87.58**	87.35	86.56	86.57
ElectG	**88.62**	88.02	88.21	88.02	88.19
Fire	84.35	83.99	**85.09**	84.28	83.66
Collision	82.49	83.66	**85.58**	84.67	85.00
Rollover	80.61	79.76	79.96	79.47	**81.51**
CollRoll	86.55	88.46	88.83	89.15	**90.09**
MiscInc	83.46	83.46	85.21	85.58	**86.55**
ShovFP	88.46	85.71	88.52	88.71	**89.63**
MovieReviews	71.44	69.12	72.12	71.47	**72.96**

Table B.1: Comparison of text classification accuracy using kNN with varying values of k.

Appendix C

Datasets and Constituent Classes

In Table C.1, we present that datasets used in this thesis and the classes that constitute each dataset.

BactV	C01 Bacterial Infections and Mycoses, C02 Virus Diseases
CardR	C14 Cardiovascular Diseases, C08 Respiratory Tract Diseases
NervI	C10 Nervous System Diseases, C20 Immunologic Diseases
MouthJ	C07 Stomatognathic Diseases, C09 Otorhinolaryngologic Diseases
NeopE	C04 Neoplasms, C21 Disorders of Environmental Origin
DigNut	C06 Digestive System Diseases, C18 Nutritional and Metabolic Diseases
MuscleS	C05 Musculoskeletal Diseases, C17 Skin and Connective Tissue Diseases
EndoH	C19 Endocrine Diseases, C15 Hemic and Lymphatic Diseases
MaleF	C12 Urologic and Male Genital Diseases, C13 Female Genital Diseases
ImmunoV	C20 Immunologic Diseases, C02 Virus Diseases
NervM	C10 Nervous System Diseases, C05 Musculoskeletal Diseases
RespENT	C08 Respiratory Tract Diseases, C09 Otorhinolaryngologic Diseases
Ohsumed01	C04 Neoplasms, C05 Musculoskeletal Diseases, C02 Virus Diseases, C01 Bacterial Infections and Mycoses, C05 Musculoskeletal Diseases
Ohsumed02	C08 Respiratory Tract Diseases, C06 Digestive System Diseases, C09 Otorhinolaryngologic Diseases, C07 Stomatognathic Diseases, C10 Nervous System Diseases
Ohsumed03	C15 Hemic and Lymphatic Diseases, C11 Eye Diseases, C14 Cardiovascular Diseases, C12 Urologic and Male Genital Diseases, C13 Female Genital Diseases

Ohsumed04	C17 Skin and Connective Tissue Diseases, C04 Neoplasms, C21 Disorders of Environmental Origin, C22 Animal Diseases, C20 Immunologic Diseases, C19 Endocrine Diseases, C18 Nutritional and Metabolic Diseases
Hardw	comp.sys.ibm.pc.hardware, comp.sys.mac.hardware
MedSp	sci.med, sci.space
CryptE	sci.crypt, sci.electronics
ChrisM	soc.religion.christian, talk.religion.misc
MeastM	talk.politics.mideast, talk.politics.misc
GunsM	talk.politics.guns, talk.politics.misc
AutoC	rec.autos, rec.motorcycles
Science	sci.crypt, sci.electronics, sci.med, sci.space
StratM	C11 Strategy/Plans, C41 Management
EntTour	GENT Arts/Culture/Entertainment, GTOUR Travel and Tourism
EqtyB	M11 Equity Markets, M12 Bond Markets
FundA	C17 Funding/Capital, C181 Mergers/Acquisitions
InRelD	GDIP International Relations, GDEF Defence
NProdRes	C22 New Products/Services, C23 Research/Development
ProdNP	C21 Production/Services, C22 New Products/Services
OilGas	I1300002 Crude Oil Exploration, I1300013 Natural Gas Exploration
ElectG	I161 Electricity Production, I162 Gas Production
Fire	Fire Injury, Fire No Injury
Collision	Collision Injury, Collision No Injury
Rollover	Rollover Injury, Rollover No Injury
CollRoll	Collision, Rollover
MiscInc	Misc Incidents Injury, Misc Incidents No Injury
ShovFP	Shovel, Fixed Plant

Table C.1: Datasets and their constituent classes.

Appendix D

Case-Based Prediction Attribute Values

This section provides the values of the attributes used for the case-based prediction framework presented in Chapter 3. This information is presented in Table D.1. All values are normalised to lie between 0 and 1.

Dataset	AveTermCount	MaxDF	AveDF	MaxIDF	AveIDF	NNSim	AveNSim	MaxNSim	MinNSim
BactV	0.03878	0.55567	0.03890	0.74036	0.54000	0.511766	0.36337	0.66004	0.13521
CardR	0.03970	0.65465	0.03974	0.69791	0.52187	0.490338	0.33837	0.64553	0.10905
NervI	0.03027	0.27520	0.03051	0.76645	0.55309	0.496093	0.32501	0.71484	0.08604
MouthJ	0.02531	0.34034	0.02533	0.79885	0.58707	0.469071	0.33009	0.68139	0.07726
NeopE	0.03505	0.60481	0.03516	0.76662	0.53468	0.462901	0.29571	0.59804	0.06974
DigNut	0.04444	0.61800	0.04444	0.71743	0.50879	0.468701	0.31423	0.57992	0.09801
MuscS	0.02832	0.29045	0.02846	0.73970	0.56985	0.496735	0.32944	0.71108	0.06285
EndoH	0.03751	0.67400	0.03751	0.74004	0.53564	0.493771	0.33827	0.64491	0.11213
MaleF	0.03225	0.58500	0.03225	0.79931	0.56702	0.501895	0.34041	0.64082	0.10009
PregN	0.03346	0.24725	0.03349	0.76640	0.54786	0.474282	0.29561	0.62667	0.05736
ImmunoV	0.03307	0.57214	0.03314	0.84062	0.57698	0.533869	0.40639	0.73867	0.16161
NervM	0.02376	0.62903	0.02395	0.84048	0.59681	0.523054	0.38849	0.76575	0.11238
RespENT	0.03354	0.60961	0.03357	0.84094	0.55475	0.492434	0.32927	0.63999	0.09897
HardW	0.03193	0.26687	0.03264	0.79837	0.56618	0.521918	0.32329	0.77404	0.06132
MedSp	0.04304	0.29382	0.04361	0.68043	0.48164	0.566539	0.40747	0.83515	0.17516
CryptE	0.05946	0.59919	0.06018	0.63905	0.44107	0.525958	0.36279	0.79737	0.11681
ChrisM	0.04885	0.72121	0.04934	0.74024	0.51435	0.600038	0.46272	0.80011	0.23195
MeastM	0.05662	0.22571	0.05731	0.65093	0.44231	0.57028	0.41933	0.83851	0.18378
GunsM	0.04501	0.65147	0.04561	0.69796	0.49259	0.572121	0.42087	0.81451	0.17447
AutoC	0.03272	0.64670	0.03321	0.71694	0.54483	0.582893	0.42273	0.82276	0.15667
StratM	0.06356	0.38138	0.06362	0.66531	0.44729	0.487876	0.35025	0.73836	0.14149
EntTour	0.07830	0.53668	0.07870	0.61753	0.40214	0.505332	0.39278	0.84895	0.19236
EqtyB	0.08592	0.68410	0.08812	0.71697	0.42311	0.599387	0.48596	0.86246	0.29282
FundA	0.07105	0.57472	0.07126	0.69811	0.43294	0.500481	0.36960	0.78150	0.16810
InRelD	0.07493	0.47400	0.07493	0.62694	0.42253	0.544525	0.42574	0.79976	0.22176
NProdRes	0.06110	0.38700	0.06110	0.68177	0.45021	0.493444	0.36170	0.75439	0.14788
ProdNP	0.06367	0.46579	0.06405	0.68106	0.43928	0.506856	0.38258	0.76416	0.17178
OilGas	0.04964	0.55946	0.05133	0.73920	0.49180	0.590904	0.46046	0.78558	0.24135
ElectGas	0.04449	0.60063	0.04639	0.67922	0.52281	0.597361	0.47670	0.83336	0.24210
Fire0	0.04705	0.84000	0.04705	0.79075	0.64875	0.43062	0.33178	0.58789	0.14211
Collision	0.05214	0.72500	0.05214	0.79075	0.63710	0.289806	0.26282	0.51077	0.07129
Rollover	0.05659	0.61000	0.05659	0.79075	0.62472	0.42871	0.30442	0.58290	0.09825
CollRoll	0.05189	0.56500	0.05189	0.79075	0.63620	0.394291	0.25973	0.53240	0.06541
MiscInc	0.03168	0.21500	0.03168	0.79075	0.68576	0.299984	0.15155	0.42739	0.01189
CraneFP	0.03684	0.52571	0.04191	0.78618	0.65962	0.319434	0.16504	0.66217	0.01053
ShovFP	0.03567	0.35227	0.04030	0.78531	0.66389	0.327621	0.17659	0.62884	0.00983
MovieRev	0.08029	0.71000	0.08029	0.74004	0.44705	0.44846	0.34567	0.53323	0.19168

Table D.1: Case-based prediction framework attribute values.