Universiteit Leiden

ICT in Business and the Public Sector

Reducing manual labor in Technology-Assisted Review

Name:	Thomas Prikkel
Date:	September 3, 2018
1st supervisor:	S. Verberne
2nd supervisor:	J.C. Scholtes

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Due to the growing amount of data stored by organisations, the cost of the eDiscovery process of legal proceedings and government investigations has increased significantly. Technology-Assisted Review (TAR) is used to reduce the cost of eDiscovery and speed up the process. In eDiscovery, achieving high recall is paramount and TAR protocols are in place to ensure defensibility of the TAR process in court. This research focuses on minimising the amount of manual labour performed by lawyers and paralegals during the TAR process while maintaining the certainty of reaching high recall on a par with the current state-of-the-art TAR standards and protocols.

We have identified three methods for reducing the amount of manual labour: (1) reducing the size of the problem, (2) increasing the return set precision and (3) changing the protocol by which the documents are labelled. The Reuters RCV1-v2 data set and a reviewed client data set are used to simulate the TAR process. Within this TAR simulation, the performance of the methods is evaluated by on the return set precision and the amount of manual labour necessary to reach a predefined recall.

We introduce a novel method to reduce the size of the problem called Topic Model-Based Filtering which utilises the cosine distances of the responsive documents in the initial training set to filter the documents in the corpus furthest away from the centroid of these responsive documents. We implement existing Machine Learning techniques to increase the precision of the return set and we introduce a novel labelling technique called Sampled Labelling, which is an extension of the current standard TAR protocol Continuous Active Learning. Sampled Labelling uses sampling to skip labelling when the precision is above a predefined threshold.

Results show that applying these techniques can reduce the necessary amount of manual labour significantly. The reduction in manual labour ranges from 10.6% to 96.9% depending on the difficulty of the task.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the last few years, the amount of data stored by companies and organisations has increased exponentially [23]. Corporate data such as email correspondence and corporate documents are highly unstructured and consist predominantly of natural language. Text mining techniques can be used to extract knowledge from these textual documents. Within the area of text mining, text classification focuses on assigning textual documents to one or more classes or categories.

The increase in corporate data can be of significant value, when effectively leveraged to provide benefits for the company. However, data can also become a liability when not stored the right way. Research shows that the risk of data loss is particularly a concern in the case of electronic discovery (eDiscovery) [38]. eDiscovery refers to the process used by organisations to deliver data such as email correspondence and other corporate documents to opposing counsel or investigators as part of legal proceedings such as litigation, acquisitions and mergers or government investigations. Such a legal request for documents and information is called a *request for production*. During eDiscovery, lawyers and paralegals conduct legal review by reviewing the documents to decide whether the documents are responsive to any of the eDiscovery requests, or should be withheld because they are confidential or privileged. Due to the increasing amount of electronically stored information, this process can be time consuming and therefore, costly. To decrease the costs and duration of legal review, Technology-Assisted Review is used.

Technology-Assisted Review (TAR) is the process of having computer software classify documents based on the contents of the documents and input from expert reviewers, in an effort to expedite the organisation and prioritisation of the document collection and review [9]. TAR assists reviewers during Issue Coding, the process in which reviewers evaluate the content of document to determine whether it relates to topics of interest in the legal proceeding. When a document is relevant for a certain case issue, the reviewers tag the document as responsive. Issue Coding is part of the legal review process. Research has shown that the use of TAR can be more accurate and time efficient than exhaustive manual review by humans [11].

Current TAR standards are to use machine learning techniques to predict which documents are responsive and which are not responsive based on the decisions applied by an expert reviewer to a small sample of documents. An often used term to describe the use machine learning to predict the issue codes for a document is *Predictive Coding*.

In this research we seek to minimise the amount of manual labour performed during the TAR process. We introduce a novel labelling technique that can significantly reduce the amount of labour necessary. Furthermore, we perform experiments on a number of methods to increase the precision of the presented documents.

1.1 Problem statement

The process of reviewing the documents is time consuming and costly, therefore automation is a popular method to reduce the time and costs of the eDiscovery process. A decrease in amount of labour performed during TAR can save a significant sum of money and could speed up this time-consuming process.

Strict protocols are in place to ensure the use of TAR is defensible in court and during preliminary meetings about the request for production. The defensibility of TAR is highly dependent on the certainty with which the documents are labelled as responsive and non-responsive. Changing the protocol to an alternative with lower certainty, could cause the change not to be implemented in real cases, even though it might reduce labour significantly. Therefore, these strict protocols restrict the possibilities of reducing manual labelling. Machine learning methods that introduce a more approximative and probabilistic approach, must be validated to guarantee performance on a par with current standards.

In investigative and legal applications, high recall is of great significance. Hence, most research in TAR is focused on increasing the recall and robustness of the learners. A lot of research is done in the field of text classification [32]. Research focusing specifically on reducing the manual effort in eDiscovery is limited.

The amount of effort necessary during legal review is determined during the *Rules of Engagement* meeting. The *Rules of Engagement* meeting is a meeting between the counsel for the defendants and the opposing counsel or investigator in which the protocol and workflow of TAR are determined and rules are discussed. One of the rules that is discussed, is the stopping criteria of legal review. An example of a stopping criteria is when the classifier labelling the documents reaches 80% precision and recall on a validation set. Even though finding all responsive documents is the objective, the precision of the classifier is important to ensure a limited number of non-responsive documents are added to the production set.

The goal of this research is to develop and evaluate methods to reduce the manual labour needed to reach the stopping criteria while maintaining the necessary degree of certainty to be defensible in court and usable in real-world applications.

1.2 Research questions

In order to find a solution for the general problem statement, three research questions were defined. The first research question focuses on identifying methods to reduce the amount of manual labour needed during legal review.

How can the necessary amount of manual labour during legal review be reduced?

After methods that could potentially reduce the amount of labour are identified, we will further investigate these methods in order to answer research question two, which focuses on the performance enhancement.

What is the effect of the identified methods to reduce the amount of manual labour?

To test the performance of the methods, we use two data sets, Reuters RCV1 and a reviewed client data set. The focus of the final research question is to investigate whether the Reuters RCV1 corpus is a suitable data set to test the performance of TAR. The performance of the experiments using Reuters data will be compared to the performance of the experiments using client data. For future research it is relevant to know whether the Reuters data set is capable of simulating a TAR process comparable to a real life case.

> *To what extent does the performance of the TAR simulation using the Reuters RCV1 corpus reflect the performance of the TAR simulation using client data?*

1.3 Structure of thesis

The structure of this thesis is as follows. Chapter 2 provides the theoretical background for Document Classification in general and TAR more specifically. In Chapter 3, all methods and techniques used in our experiments are described and the two data sets used in the experiments are introduced. Any necessary information to implement and reproduce this work is also provided in the aforementioned chapter. The experimental setup consists of three separate experiments that are performed within the TAR simulation and is described in detail in Chapter 4. After describing all the experiments, the results of these experiments are presented in Chapter 5. In Chapter 6, we elaborate on the results and discuss our findings. In the final chapter, we answer the research questions, share our conclusions and specify what we would consider feasible and valuable future work.

Chapter 2

Background

In the background, an overview of the field of research is provided and relevant previous work is discussed. First, text classification in general is discussed and afterwards the application of text classification in TAR.

2.1 Binary text classification

The problem as described in the problem statement is a binary text classification problem. The goal of text classification, which is also know as text categorisation, is to classify documents into a fixed number of predefined classes. Each document can be classified in either multiple, exactly one, or no category at all. Binary text classification is a distinctive type of single-label text classification, where every document must be assigned either to class c_i or to its complement $\overline{c}_i$.

2.2 Document Representation

In order for classifiers to be able to work with text documents, the data needs to be converted to a suitable representation. A common way to represent a document d_j in the field of text classification is as a vector of term weights $d_j = \{w_{1j}, \ldots, w_{|T|j}\}$ where T is set of terms. There are numerous different approaches to calculate this term feature vector since there are different ways to define a term and different ways to compute the term weights. A collection of documents, also known as a corpus, is represented by a document-term matrix where each row is a term feature vector for a document. Feature vectors where the terms represent the words in the documents are most commonly used. Using this document representation, the word order of the text is lost and is therefore called Bag-of-Words. Another type of terms are N-grams, where each term is represented by a combination of N words, this is used to store spatial information and maintain some sense of the word order within the text. A straightforward example of term weight is *Term Frequency*, counting all the occurrences within a document.

2.2.1 TF-IDF

The Term Frequency Inverse Document Frequency (TF-IDF) feature vector representation identifies terms with words. TF-IDF is an extension of the regular Bag-of-Words approach, which solely counts the occurrences of each term in the documents. TF-IDF also incorporates how often it occurs in all other documents. TF-IDF embodies the intuitions that (1) the more often a term occurs in a document, the more it is representative of its content, and (2) the more documents the term occurs in, the less discriminating it is [32]. The function for determining the TF-IDF weight $w_{k,j}$ of term k in document j is as follows [30]:

$$w_{k,j} = tf_{k,j} \cdot log\frac{N}{df_k}$$

where $tf_{k,j}$ denotes the term frequency, the number of times term k occurs in document j, and df_k denotes the document frequency, the total number of documents N in which term k occurs. A log function is used to ensure terms with a high IDF value are not disproportionately boosting the document scores and to stop a floating point underflow from occurring [28]. Floating point underflow occurs when a float is so small that the value can no longer be stored properly.

2.3 Learning scenarios

Throughout this thesis, three distinct learning scenarios are used. All three learning scenarios are described below.

2.3.1 Supervised learning

In supervised learning, the training data consists of an input object and a desired output. Hence, the data is labelled. The labelled data is used to train an algorithm which subsequently makes predictions for all unseen data. This learning scenario is associated with classification, regression and ranking problems [24]. The downside of using this learning scenario, is that it requires a lot of manual labour to label the data required to effectively train the learner.

2.3.2 Semi-supervised learning

When semi-supervised learning is used, the learner receives a training set containing both labelled and unlabelled data. Semi-supervised learning is a common scenario in problems where unlabelled data is easily accessible but labelled data is expensive to obtain [24]. Various applications can be framed as instances of semi-supervised learning, such as classification, regression and ranking. Compared to supervised learning, this learning scenario makes use of unlabelled data to learn and therefore requires less labelled data. Hence, less manual labour is required to create the training data.

2.3.3 Unsupervised learning

In the unsupervised learning scenario, the algorithm receives exclusively unlabelled data and makes predictions about the unlabelled data. Given that no labels are provided, it is often difficult to quantitatively evaluate the performance of an unsupervised learner. Clustering and dimension reduction are problems associated with unsupervised learning [24]. The advantage of using unsupervised learning is that no additional labelling is required.

2.4 Technology-Assisted Review

To reduce the time needed for experts to review the documents involved in an eDiscovery request, TAR is used. The TAR process starts with a set of documents and request to produce a set of responsive documents. The request for production one or more *issues* that define the topic and scope of the documents that should be labelled as responsive. An increasing amount of tools are available to the human operator to identify documents that should be shown to one or more human reviewers. Traditionally, boolean search was one of the common tools used. A disadvantage of boolean search is that the returned result is all or nothing. Therefore, in recent years, a shift has been made to more sophisticated machine learning techniques to help the human operator determine the next batch of documents to be reviewed.

The reviewers examine the documents that are served to them and label ("code") them as responsive to the issue or not. Iteratively, more documents to be labelled are identified using the tools and in turn examined by the reviewers. This process continues until enough of the responsive documents have been reviewed and coded. The definition of 'enough' is determined by lawyers during the production request meetings. 'Enough' is often based on how much additional effort it would likely take to find more responsive documents and how important those documents will be in resolving the legal dispute [5]. A strict protocol is followed during the TAR process to ensure the validity and defensibility of the TAR process. A number of different protocols can be used to find the responsive documents, which are described in the next section.

Research has shown that exhaustive manual review, where the reviewers go through all the documents and manually label them, is not always the most effective approach [11]. Grossman and Cormack list a number of studies where inconsistencies were found between two or more teams reviewing the same documents [11]. The question is raised as to whether there is a gold standard in legal review. In the studies, different approaches to resolve this problem are followed. In one study [42], the primary assessor composed the request for production and is therefore deemed the gold standard. In another study, documents lacking consensus were reviewed by a senior litigator, who decided which team had made the correct decision [29]. Aside from the inconsistencies in the manual review, the performance of the manual review in terms of F1-score was inferior to the performance of review using TAR.

A study of popular eDiscovery algorithms which compared the performance of Logistic regression, Linear SVM, Gradient Boosting, Multi-layered Perceptron and 1-Nearest Neighbour found that Linear SVM outperforms all other methods [43]. The study also compares three different types of document representations: (1) Bag-of-words, (2) Term frequency and (3) TF-IDF. TF-IDF is the superior document representation to use with Linear SVM in terms of classification performance.

Given that humans make mistakes when labelling the documents [8], a study investigated what the effect of errors in the training data is on the performance of the classifier [31]. By injecting up to 25% erroneous training documents, they found that, for 25% incorrect training documents, the loss in F1-score ranges from 3.2% to 5.3%. The study also researched the effect of rolling collections on the performance of the classifier. In legal and investigative applications, it is often the case that not all data is available from the start. Meaning that new data is constantly added to the collection, which is called a rolling collection. This leads to a document representation that does not take all features of the entire document collection into consideration. The study found that rolling collections cause a significant drop in performance and it is recommended to recalculate, train and verify the entire machine learning model on the entire document collection for every addition of new documents [31].

The effect of errors in the training data was also researched within a TAR simulation in another study and finds that within the simulated TAR process, the loss in performance is also limited [36]. The study investigated what the effect is of circumstances that may negatively affect the performance of TAR such as human review error, difference in document length and class imbalance. The study looks at approaches to counter these influences on performance.

The TREC 2015 Total Recall Track focused on methods designed to achieve very high recall with a human assessor in the loop [10]. Beating the baseline turned out to be difficult [40]. The baseline was created by two program coordinators, who implemented a continuous activate learning solution in combination with logistic regression [4].

Effective similar document detection can dramatically decrease the costs of the TAR process, as the number of similar documents in typical electronic discovery corpus ranges between 25% and 50% [37]. The study investigates the use of similar document detection in eDiscovery and introduces a novel algorithm to detect similar documents [37].

A paper published at the 2015 DESI workshop discussed the use of statistical sampling to approximate the number of relevant documents in a corpus and perform quality control on the predictive coding results [26]. This paper also states that the use of TAR versus traditional linear review could cut costs by 30% and save 74% of the hours spent on review.

Another paper published at the 2015 DESI workshop researched the effect of including metadata in the machine learning process on the performance of TAR [16]. Multiple ways of including the metadata in the machine learning process were investigated. The best performing method for including the metadata is to create two separate models, one for the metadata and one for the text, and combine them afterwards. This method is sometimes called *Late Fusion* [3]. The results show that all available metadata can increase performance significantly [16].

2.5 TAR protocols

In this section, the different protocols for the TAR process are described. Research has shown that Continuous Active Learning achieves the highest performance of the three protocols [5]. Table 2.1 shows an overview of the main characteristics of the TAR protocols discussed below.

	Initial training set	**New training documents**
SPL	random	random
SAL	search query or random	least certain
CAL	search query	top scoring

Table 2.1: Main characteristics of TAR protocols [36]

2.5.1 Simple Passive Learning

The Simple Passive Learning (SPL) protocol is dependent on the operator or random selection, and not the learning algorithm, to identify the training set. The operator is a person appointed to operate the TAR software and may or may not be a member of the team of reviewers. The operator can use tools such as boolean search to identify and select documents that will be part of the initial training set (seed set). The initial training set contains both responsive and non-responsive documents. The candidate training set is used to train a classifier and the classifier is in turn used to generate a candidate review set from the unlabelled set. If the candidate review set is of "inadequate" quality, the operator creates a new candidate training set, generally by adding new documents that are found by the operator, or using random selection. This process continues until the quality of the review set is adequate and the review set is served to and coded by the reviewers. SPL is outperformed by Simple Active Learning and Continuous Active Learning [5].

2.5.2 Simple Active Learning

The Simple Active Learning (SAL) protocol begins with the creation of an initial training set. The initial training set may be selected using keyword search, random selection or both. Same as SPL, the initial training set contains both responsive and non-responsive documents. The initial training set is used to train a learning algorithm which is to compute a probability to be responsive for each of the unlabelled documents. The review set is selected using *uncertainty sampling*, a method that selects the documents about which the classifier is least certain. The review set is reviewed, coded and added to the training set. This iterative process continues until the cost of reviewing and coding more documents outweighs the benefit of adding more documents to the training set. This point is often referred to as "stabilisation" in the context of TAR. At this point, the classifier is used a final time to create a set or ranking of likely relevant documents, which are manually labelled. Uncertainty sampling shares a fundamental weakness with passive learning: the need to define and detect when stabilisation has occurred [5]. Stopping early could result in insufficient recall.

2.5.3 Continuous Active Learning

The Continuous Active Learning (CAL) protocol consists of a keyword search system and a learning algorithm. The operator typically uses a keyword search to select the initial training set to be reviewed and coded. Just like SAL, the initial training set is used to train a learning algorithm which is used to assign each document with a probability to be responsive. Unlike SAL, the documents with the highest probabilities to be responsive are reviewed and coded. Reviewing and coding is done in batches, after which a new classifier is train. These newly reviewed documents are added to the training set of the next iteration. This iterative process continues until 'enough' of the responsive documents have been found. Research has shown that CAL achieves superior performance compared to SAL and SPL [5]. An example of a stopping criteria for CAL could be to continue until the return set precision has dropped below 5%.

Chapter 3

Methodology

To reduce manual labour in the TAR process we have implemented a number of techniques. All methods and techniques we have implemented are described in this chapter. There are multiple methods for reducing the amount of manual labelling needed to reach the recall as defined in the rules of engagement meetings. The strategy to reduce manual labour is highly dependent on the TAR protocol. Given that the CAL protocol is the current state-of-the-art TAR protocol [5], we focused on reducing the manual labour using this protocol as the baseline.

The first identified method to reduce the amount of labour necessary to reach the predetermined recall is reducing the size of the problem. Reducing the size of the problem prior to starting the TAR process could potentially decrease the required labelling effort.

The CAL protocol requires all the documents that are labelled as responsive to be manually labelled. Therefore, in order to reduce the number of documents that have to be manually labelled, the return set precision must be increased. Increasing the return set precision will allow the legal expert to reach the needed recall faster because less non-responsive documents have to be manually labelled. Multiple approaches to increase the precision of the return set are described in Section 3.3.

As an extension to the CAL protocol, we present a novel labelling method that reduces the amount of labelling effort required while preserving the certainty that no documents are incorrectly labelled as non-responsive. Hence, diminishing the disadvantage of CAL which requires all responsive documents to be manually labelled.

Figure 3.1 portrays the different methods explained in this section and how they were used in the three different experiments which are described in the next chapter. The figure also shows how the experiments are the input for one another. The Topic Model-Based Filtering results were used in the 'Increasing Return Set Precision' experiment and the highest scoring component from that experiment was used in the Sampled Labelling experiment.

This chapter starts with a description of the baseline algorithm. In the following sections, the algorithms used in the three experiments are described. Afterwards, we the data sets used to conduct the experiments are elaborated on. The last section of the chapter provides any necessary information about the implementation of the algorithms to reproduce the experiments.

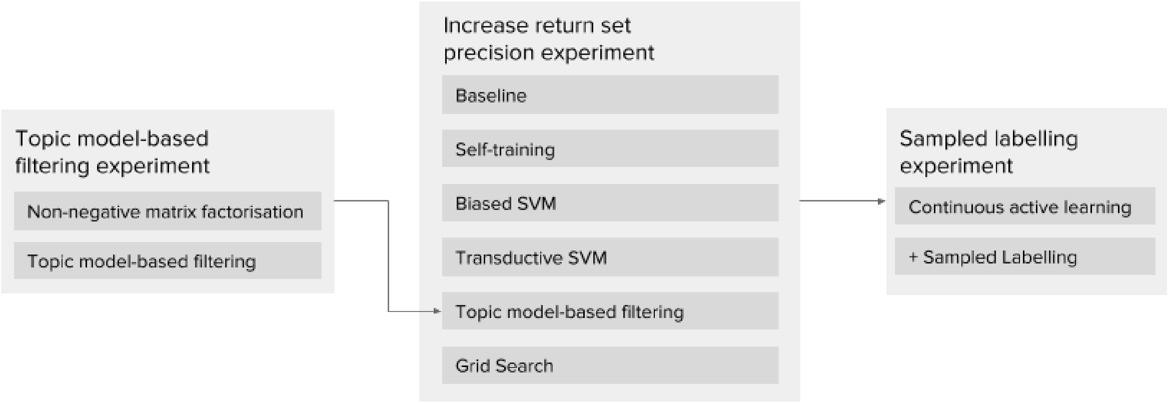
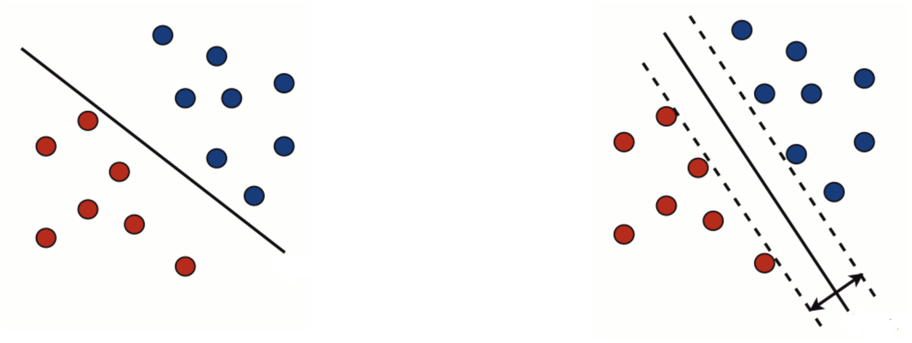

Topic model-based filtering experiment
Non-negative matrix factorisation
Topic model-based filtering
Increase return set precision experiment
Baseline
Self-training
Biased SVM
Transductive SVM
Topic model-based filtering
Grid Search
Sampled labelling experiment
Continuous active learning
+ Sampled Labelling

Figure 3.1: Overview of experiments and methods within the TAR simulation.

3.1 Baseline

The baseline implementation used in this research is a standard Linear Support Vector Machine without any parameter tuning. Support Vector Machines (SVMs) were first introduced by Vapnik et al. [41]. SVMs are supervised and require labelled training data. Jaochims first explored the benefits of SVMs for text classification [15]. SVMs are one of the most theoretically substantiated classification algorithms in modern machine learning [24]. The goal of a SVM is to find a hyperplane that separates the training set into a positive and a negative labelled set. From all possible hyperplanes, the SVM seeks to find the hyperplane with the maximum *margin*, or distance to the closest points, and is thus known as the *maximum-margin hyperplane* illustrated in Figure 3.2b.



(a) Possible hyperplane

(b) Hyperplane that maximises margin

Figure 3.2: Two possible separating hyperplanes [24]

Given a training set of l instance label pairs (x_i,y_i) with labels $y_i \in \{-1,+1\}$ and x_i as feature vectors for $i = 1,...,l$. SVMs try to find a maximum-margin hyperplane. The hyperplane is given by:

$$w \cdot x + b = 0 \tag{3.1}$$

where $w \in \mathbb{R}^N$ is the weight vector normal to the hyperplane and $b \in \mathbb{R}$ is a scalar. The maximum-margin hyperplane can be found by solving an optimisation problem. When the data is not fully linearly separable, slack variables can be introduced to allow a number of misclassified instances and still be able to find a maximum-margin hyperplane. This is referred to as *soft margin*, as opposed to *hard margin* in the linearly separable case. The optimisation problem is formulated as follows

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{l} \xi_i^p \\ \text{subject to} \quad & y_i(w \cdot x_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0,\ i = 1,\ldots,l \end{aligned} \tag{3.2}$$

where ξ_i are slack variables and the second term is a regularisation term with the parameter C that controls the insignificance of misclassification. A slack variable ξ_i measures the distance by which vector x_i violates the desired inequality, $y_i(w \cdot x_i + b) \geq 1$. The size of the penalty for slack variables is determined by p. The values $p = 1$ and $p = 2$ lead to the most straightforward solutions and analyses and are called *hinge loss* and *quadratic hinge loss*, respectively. Hinge loss is the most widely used loss function for SVMs.

The optimisation problem in combination with the constraints can be solved using the method of Lagrange multipliers. The Lagrange multiplier uses the formula: $L(\alpha, X) = f(X) + \alpha g(X)$ where $f(X)$ is the optimisation condition that is subject to the constraint $g(X)$. In the case of the aforementioned optimisation problem, the dual form is formulated as follows:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\ \text{subject to} \quad & \sum_{i,j=1}^{l} \alpha_i y_i = 0, 0 \leq \alpha_i \leq C,\ i = 1,\ldots,l \end{aligned} \tag{3.3}$$

Solving equation 3.3 and finding the optimal values for α_i, it can be written that:

$$w = \sum_{i=1}^{l} \alpha_i y_i x_i \tag{3.4}$$

Most of the weights w_i will be zero, only the support vectors will have nonzero weights and determine the position of the hyperplane. The solution α of the dual problem (3.3) can be used directly to determine the hypothesis returned by SVMs:

$$h(x) = sgn(w \cdot x + b) = sgn(\sum_{i=1}^{l} \alpha_i y_i (x_i \cdot x) + b) \tag{3.5}$$

The hypothesis solution depends solely on the inner products between vectors and not directly on the vectors themselves. This characteristic can be used to extend SVMs to find non-linear decision boundaries using *kernels*.

One remarkable property of SVMs is that their ability to learn is independent of the dimensionality of the feature space [15] making them highly suitable to work with high dimensional input data. Additionaly, Joachims proved there are few irrelevant features in text classification and demonstrates that most text classification problems are linearly separable. Furthermore, he shows that SVMs perform well with sparse document vectors [15].

The baseline implementation does not use any parameter optimisation and therefore uses the default parameters and weights. The default weight for the C parameter is 1 for both the positive and negative class.

3.2 Topic Modelling

The first labour reduction method is based on Topic Modelling. By significantly reducing the feature space, distances between documents can be calculated relatively quick and without a lot of processing power. These distances are utilised to determine the responsiveness of the documents.

Nonnegative Matrix Factorisation (NMF) is used to create the topic models for the TF-IDF matrix of the full corpus. The goal of NMF is reducing the dimensionality of the data. NMF is an unsupervised learner that can be used to cluster documents. Contrary to alternative dimensionality reduction methods, such as LDA and PCA, NMF does not allow negative factorisation. The non-negative constraint is especially important for textual data, since terms cannot have a negative number of occurrences or a negative TF-IDF value. Adding the nonnegative constraint makes NMF particularly suitable for clustering textual data[1].

NMF is used to compute a lower rank approximation of a large sparse matrix by clustering documents on the basis of shared semantic features [33]. NMF is used as a technique for document clustering and topic modelling [17]. Given a data matrix V of dimensions $N \times F$ with nonnegative entries, NMF is the problem of finding a factorisation

$$V \approx WH \tag{3.6}$$

where W and H are nonnegative matrices of dimensions $N \times K$ and $K \times F$ [7]. K is generally chosen so that $(N + F)K \ll FN$ [17]. Figure 3.3 illustrates how matrix V is decomposed into matrix W and H. The matrix V is a document-term matrix with N documents and F TF-IDF features. With a chosen K, factorisation of matrix V results in a document-topic matrix W with N documents and K topic weights for each of the documents. Hence, in the document-topic matrix the weights with which a document belongs to the topics are stored.

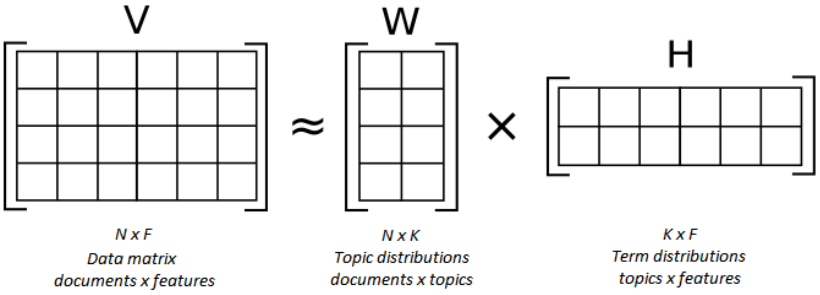


Figure 3.3: NMF: The factorisation of matrix V in Matrix W and H[12]

NMF differs from other rank reduction methods due to the use of constraints that produce nonnegative basis vectors W and H. These constraints are imperative for the success of parts-based representation [17]. Since W and H contain no negative entries, this allows only additive combinations of the vectors to reproduce the original. This implies that each document vector in the matrix V can be explained by a linear combination of the topics.

We use the responsive seed set, the set of responsive documents with which the TAR process is started, to calculate the centroid of the NMF vectors for the categories at hand. To get the centroid of the NMF vectors, the mean value for each of the topics was calculated and a new vector containing these values was created. After determining the centroid of the category, the cosine distance is calculated from each instance in the corpus to this centroid. The cosine similarity between two vectors d_i and d_j is given by:

$$cos(d_i,d_j) = \frac{d_i^T d_j}{\|d_i\|\|d_j\|} = \frac{\sum_{k=1}^{t} d_{ik} d_{jk}}{\sqrt{\sum_{k=1}^{t} d_{ik}^2}\sqrt{\sum_{k=1}^{t} d_{jk}^2}} \tag{3.7}$$

where t denotes the number of terms or features in the vector. To obtain the distance, the cosine similarity is subtracted from the value 1. It is important to note that the cosine distance is not a proper distance metric, since it does not have the triangle inequality property. For textual data, a valuable property of the cosine distance is the document length independence.

Two distance distributions are created for the corpus: (1) distance from the documents in the seed set to the centroid and (2) distance from all other documents to the centroid. All cosine distances are values between 0 and 1. These distances are the starting point for the technique described below.

3.2.1 Topic Model-Based Filtering

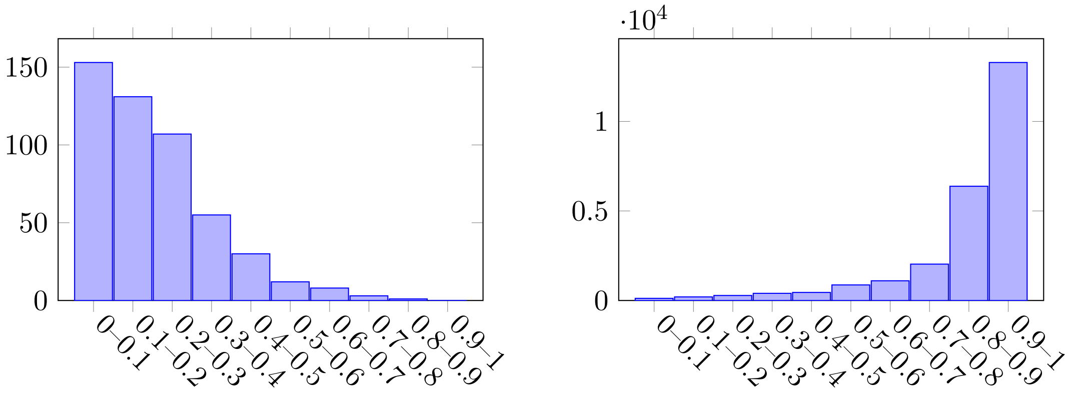
The hypothesis is, that the documents furthest away from the centroid are most likely to be non-responsive. This would allow a number of documents to be labelled as non-responsive before even starting the TAR process, thereby significantly reducing the size of the problem and increasing the size of the negative training data.

To determine the number of documents to be filtered, the following steps are executed:

1. Create two histograms by binning (1) the distances of the documents in the seed set to the centroid and (2) the distances of all other documents in the corpus to the centroid. This is illustrated by Figure 3.4.
2. After retrieving the count for each bin, all the bins are normalised to attain the percentage of documents in each bin for both histograms. This is illustrated in Figure 3.5
3. Both bins are compared starting from the bins containing the documents with the highest distances. Whenever the percentage of the bin containing the responsive documents from the seed set is lower than p percent of the percentage of the bin containing all documents, the complete bin will be filtered. p is a threshold for the trade-off between filtering a large number of documents and the certainty that not many responsive documents are filtered. This step is demonstrated in Table 3.1.

In the example of Table 3.1, where $p = 0.01$, the bins $0.8 - 0.9$ and $0.9 - 1$ are both filtered. For all the other bins, Bin 1 is not smaller than Bin 2 $*$ p.

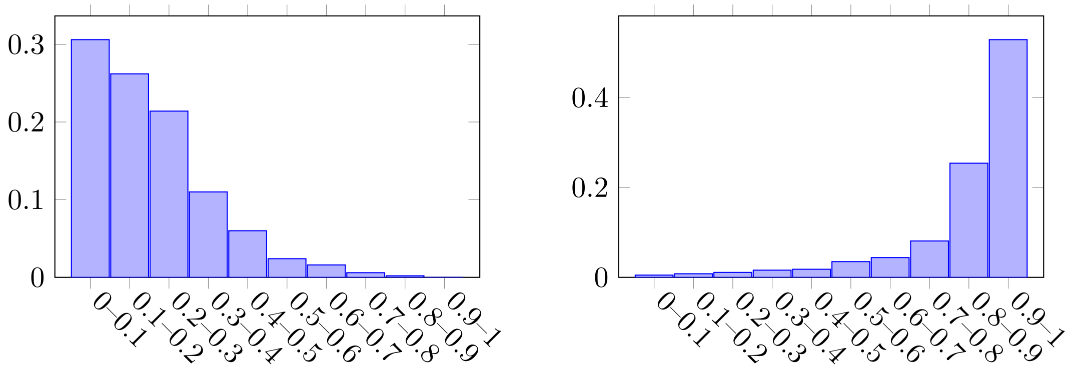
After computing the bins that have to be filtered, the documents inside the bins are automatically labelled as non-responsive. Hence, the initial seed set is enlarged with non-responsive documents. Topic Model-Based Filtering is used as a way to quickly increase the amount of training data to increase the return set precision.



(a) Distances from centroid to responsive seed documents in corpus

(b) Distances from centroid to all other documents in corpus

Figure 3.4: Distance histograms



(a) Normalised distances from centroid to responsive seed documents in corpus

(b) Normalised distances from centroid to all other documents in corpus

Figure 3.5: Normalised distance histograms

Distance	Bin 1	Bin 2	Bin 2 $* p$
0 – 0.1	0.3060	0.0048	0.0000
0.1 – 0.2	0.2620	0.0080	0.0001
0.2 – 0.3	0.2140	0.0111	0.0001
0.3 – 0.4	0.1100	0.0159	0.0002
0.4 – 0.5	0.0600	0.0179	0.0002
0.5 – 0.6	0.0240	0.0346	0.0003
0.6 – 0.7	0.0160	0.0438	0.0004
0.7 – 0.8	0.0060	0.0809	0.0008
0.8 – 0.9	**0.0020**	**0.2540**	**0.0025**
0.9 – 1	**0.0000**	**0.5291**	**0.0053**

Table 3.1: Bins for $p = 0.01$

3.3 Increase return set precision

The second method for reducing the amount of manual labour necessary during TAR is to increase the return set precision of the classifier. Five methods for increasing the return set precision were identified and described below.

One characteristic of the TAR problem is that the amount of available labelled data is limited, especially at the beginning of the review process. Therefore, it is important that the classifier is able to learn quickly from limited labelled data. The most intuitive way to achieve this is by looking for classifiers that learn using the information stored in the unlabelled data. Hence, we are switching from supervised learning to semi-supervised learning.

3.3.1 Self-training

To increase the batch size precision and reaching a higher recall faster, we implement a wrapper-algorithm that utilises self-training. A wrapper-algorithm is an algorithm that is uses another algorithm as basis to establish what the possible class for each document is. The idea is to reduce reviewing effort by training on unlabelled data. Self-training is one of the earliest ideas about using unlabelled data in classification [25]. Only labelled data is used to train a supervised method in the first iteration. The supervised method is retrained using its own prediction as additional labelled training data. Self-training can continue for multiple iterations, each iteration adding the instances with high confidence to the training data. The self-training method can be used to wrap any supervised learning method.

Algorithms using the approach of self-labelling, of which self-training is one example, can be divided into two categories [39]: (1) inductive learning and (2) transductive learning. Inductive learning is viewed as traditional supervised learning, where a model is trained using labelled training data and the main objective is to make predictions about the labels of instances that are unseen and unknown. Hence, inductive learning attempts to create generalisations based on the training data. Opposed to inductive learning, the aim of the transductive learning is to predict the true label of unlabelled instances which are also used to train the model. Instead of building a generalised model, the information contained in the unlabelled instances is leveraged to make better predictions.

Self-training is an inductive learner that shows promising results with limited amounts of labelled data. A limitation of self-labelling methods is that the performance is upper-bounded by the traditional supervised learning algorithms used as base classifiers [39].

Self-training seems to be suitable for the TAR process, since there is always limited available labelled training data and plenty of unlabelled documents.

3.3.2 Biased Support Vector Machine

The Biased Support Vector Machine (BSVM) is an alternative classifier that we implement to see whether this improves the performance. The BSVM is an approach to tackle the PU learning problem. PU learning is the task of learning from Positive and Unlabelled data. PU learning could be very useful for reducing manual labour since it changes the SVM from a supervised learning method to a semi-supervised learning method. The ability to train on unlabelled data could eliminate the necessity to label documents.

There are three distinctive methods for solving the PU learning problem: (1) using only the labelled positive examples, (2) a two-step strategy and (3) a one-step strategy. The perfect example of a classifier following the first strategy is the one-class SVM [22]. The two-step strategy iteratively goes through the following two steps. Step 1: Determine the reliable positive or negative instances from the unlabelled set to enlarge the original training set. Step 2: Building a set of classifiers by iteratively applying a classification algorithm and then selecting the best classifier from the set [20]. The one-step methods transform the PU problem into an imbalanced binary classification problem by assigning the unlabelled data to the negative class. The BSVM used in this research is an approach from the third category using a one-step strategy.

To use the BSVM, the problem is converted to an imbalance binary classification problem by supposing that the unlabelled data belongs to the negative class. Therefore, $x_1, ..., x_l$ are positive and x_{l+1},...,x_m belong to the negative class. The formula for the regular SVM (3.2) is altered to allow for two separate penalty factors of misclassification for the positive and unlabelled data. The BSVM optimisation problem is as follows.

$$\begin{array}{rl} \min\limits_{w,b,\xi} & \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C_p \sum_{i=1}^{l} \xi_i + C_n \sum_{i=l+1}^{l+m} \xi_i \\ \text{subject to} & y_i(w \cdot x_i + b) \geq 1 - \xi_i, \, \xi_i \geq 0, \, i = 1,...,l+m \end{array} \tag{3.8}$$

Where C_p and C_n are the penalty factors for the positive and unlabelled instances, respectively. Most often, the C_p is larger than C_n since the the unlabelled set also contains positive data [21].

Previous work shows that the BSVM's performance is superior to existing two-step techniques [21]. The BSVM seems to be very suitable for a problem where there is a lot of unlabelled data and labelled data is scarce. The advantage of using a BSVM versus a regular SVM is that it utilises this unlabelled data. The aim of this component is to increase the return set precision and thereby reducing the amount of manual labour needed to reach a high recall.

Since there is a large class imbalance in the training data compared to the regular SVM, the defaults weights for the positive and negative class are not equal. The negative class also contains positive instances that are labelled as negative for training purposes. The default weights for the BSVM are: $C_p = 1$ and $C_n = 0.001$. C_p should be significantly larger than C_n, this asymmetric cost formulation has been used to solve the unbalanced data problem [21]. For the experiment, we will use the default weights, parameter optimisation will be covered as a separate component which we discuss later in Section 3.3.5.

3.3.3 Transductive Support Vector Machine

We use a second alternative classifier called the Transductive Support Vector Machine (TSVM). The TSVM is a transductive learner and uses both labelled and unlabelled data to train [14]. The implementation of the TSVM we use in our experiment, as described in Section 3.6, implements large scale version of the TSVM [34]. Transductive SVM appends an additional term in the SVM objective function whose role is to drive the classification hyperplane towards low data density regions.
The TSVM optimisation problem is as follows:

$$\begin{array}{rl} \min\limits_{w,b,\xi} & \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C_p \sum_{i=1}^{l} \xi_i + C_u \sum_{i=l+1}^{l+m} \xi_i \\ \text{subject to} & y_i(w \cdot x_i + b) \geq 1 - \xi_i, \, \xi_i \geq 0, \, i = 1,...,l+m \\ & |w \cdot x_i + b| \geq 1 - \xi_i \end{array} \tag{3.9}$$

Where $x_1, ..., x_l$ are positive and x_{l+1},...,x_m are unlabelled instances. For $C_u = 0$ in Equation 3.9 we obtain the standard SVM optimisation problem. For $C_u > 0$, unlabelled data that is inside the margin is penalised. The default parameters are $C_p = 10^{-5}$ and $C_n = 10^{-5}$ [14]. The last parameter which can be passed to the algorithm is R, the fraction of the unlabelled data that should be labelled as positive. Previous work shows that the TSVM outperforms the regular SVM when classifying text [14].

3.3.4 Applied Topic Model Based-Filtering

For the third component, the result of the Topic Model-Based Filtering experiment is used to decrease the size of the problem in TAR and quickly increase the size of the training data. The filtered documents will be added to the negative class and used to train the classifier. The parameter p for the Topic model-based filtering we use in this experiment will be described in the Experimental Setup.

Our hypothesis is, that by decreasing the size of the problem and starting with more (negative) training data, the performance of the classifier will increase.

3.3.5 Grid Search

Hyperparameter optimisation is an important strategy to improve the precision of a classifier and therefore decrease the manual labour needed to reach a certain recall. To effectively tune the parameters of the classifiers, we used Grid Search in combination with a hold-out validation set. The validation set has to be manually labelled, therefore a trade-off is made between the size of the validation set and the amount of manual labour required to label the validation set.

For the parameter optimisation strategy to be effective, the performance of the classifier must increase enough to overcome the extra labelling effort to create the validation set. Therefore, the size of the validation set is very important. A larger validation set can approximate the performance on the full corpus more accurately but requires more manual labelling. The validation set should be large enough to provide an accurate indication of the performance of the classifier, but not larger than necessary.

Grid search can be used to optimise any parameters of the other components, meaning this component could be be combined with for instance the BSVM or TSVM. What the range of parameters for each classifier is, will be discussed in the next Chapter. The validation set will also give an approximation of the prevalence of the responsive documents. Which could help determine when to stop the TAR process.

3.4 Sampled Labelling

In order not to manually label all responsive documents, we have created a novel labelling method to skip parts of the labelling process while maintaining a high degree of certainty. The regular CAL protocol forces every responsive document to be manually labelled. Following this protocol yields the highest performance in terms of obtaining high recall but does require a lot of manual labour. Especially for issues where the classifiers are performing well from the start, where the return set precision is high, the labelling performed by the legal expert could be called redundant labelling effort. In order to decrease the redundant labelling effort, a method we call *Sampled Labelling* is used.

Sampled Labelling makes use of the CAL protocol property that every iteration yields the most probable documents from the return set. Instead of labelling all the top documents, a batch of documents of size *skip size* are skipped. The skipped documents are not immediately labelled as responsive, instead, they are validated by a batch of documents succeeding the initially skipped documents. A batch of the succeeding documents are manually labelled and if the precision of these documents is above a certain threshold, the skipped documents are automatically labelled as responsive. The batch of documents manually labelled to validate the skipped documents can be adjusted in size, a larger batch for greater certainty that the batch precision corresponds to the skipped documents precision. Whenever the return set precision of the manually labelled documents is above the predefined *skip threshold*, documents are skipped. Our hypothesis is that the preceding batch will also have a high precision and can therefore be labelled as responsive. When the return set precision is above a second threshold, *double threshold*, the *Skip size* is doubled. If the return set precision is not above the predefined *skip threshold*, the *skip size* is divided by two. If the *skip size* is smaller than 10% of the *batch size*, the initially skipped documents are manually labelled and no documents are skipped. When no documents are skipped, Sampled Labelling works identical to the regular CAL protocol.

The psuedocode below demonstrates how Sampled Labelling is implemented. The input of Sampled Labelling is the output of the classifier within the TAR simulation: a list of predicted probabilities for each document to be responsive. Sampled Labelling will serve the reviewer with documents to review, in batches of the size $Sample\ size$. It will continue serving the reviewer with new documents to review until $Batch\ size$ documents are reviewed and a new classifier is trained.

Sampled Labelling

```
Input:
Sorted predicted document probabilities
Batch size, Sample size, Skip size, Skip threshold, Double threshold

while $\#Labelled\ documents < Batch\ size$ do
    Select batch of $Skip\ size$ documents to skip from top of document list
    Select succeeding batch of $Sample\ size$ documents to validate result
    Review documents in sample and calculate the $Batch\ precision$
    $\#Labelled\ documents\ += \ Sample\ size$
    if $Batch\ precision > threshold$ then
        Label skipped documents as responsive
        if $Batch\ precision > Double\ threshold$ then
            $Skip\ size\ =\ Skip\ size\ *\ 2$
        end if
    else
        if $Skip\ size\ >\ (Batch\ size\ /\ 10)$ then
            $Skip\ size\ =\ max(Skip\ size/2, Batch\ size\ /\ 10)$
        else
            Review batch of skipped documents until $Batch\ size$ documents are reviewed
            $\#Labelled\ documents\ =\ Batch\ size$
        end if
    end if
end while

output:
Updated document labels
```

Figure 3.6 shows the distinction between CAL and Sampled labelling as to how the review batch is selected. CAL selects the top n documents as review batch, where n can be set to any value the users prefer. The default value for n is $1{,}000$ meaning documents are reviewed in batches of thousand documents and after every batch has been reviewed, the model is retrained. Sampled Labelling does not pick the top n documents for review but instead skips a number of documents. To validate the precision of the skipped documents, the return set precision of a sample review batch of a smaller size (e.g. $n/10$) is calculated and when this is above a certain threshold (e.g. 0.9 precision) the documents are skipped indefinitely and added to the responsive training set. Without retraining the model, the next sample review batch is selected after more documents have been skipped. This process of skipping and reviewing samples continues until n documents are manually reviewed, after which the model is retrained.

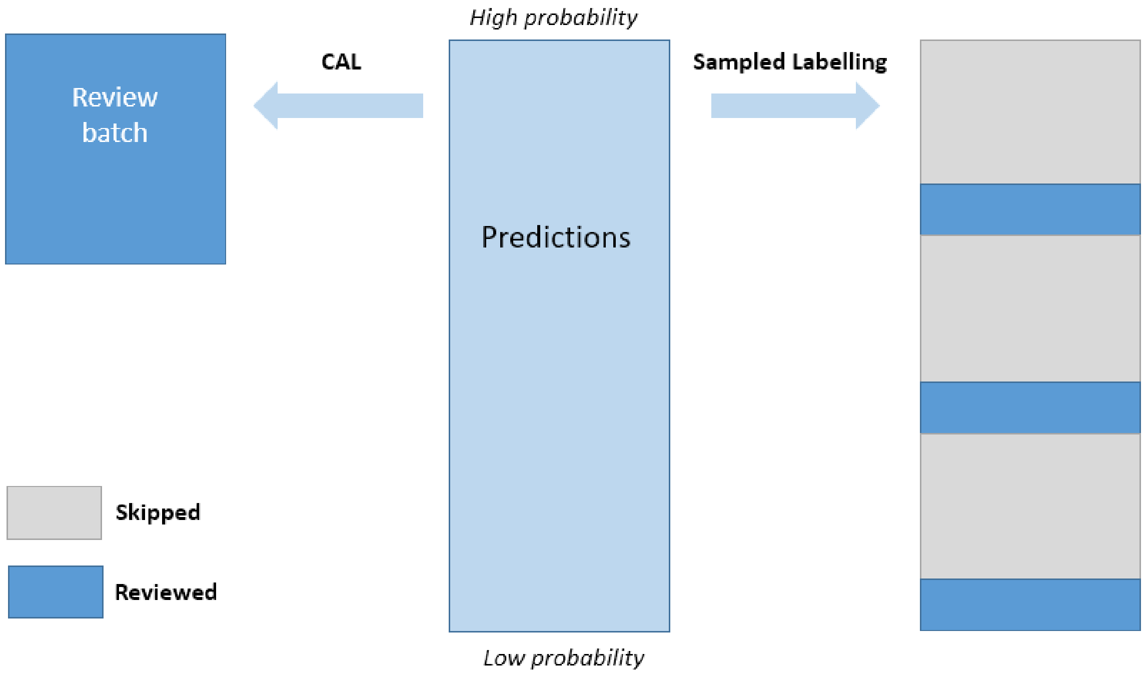


Figure 3.6: Sampled Labelling compared to CAL

Certainty about the responsiveness of the automatically labelled documents is exchanged for a labour reduction. However, this method will only allow non-responsive documents to be labelled as responsive, not vice versa. In eDiscovery, labelling non-responsive documents as responsive is less erroneous than labelling responsive documents as non-responsive [2]. This is due to the fact that after finding all the responsive documents, the documents have to be checked for privilege and non-responsive documents can be filtered there.

The number of skipped documents, the *skip size*, is in turn determined by the manually labelled batch' precision. When this precision is above a certain threshold, the skipped size is increased, doubled for instance. This technique allows for faster labelling when the precision is high. If the precision is not above the threshold to skip the documents, the skip size is decreased, divided by two for instance.

3.5 Data sets

Two data sets are used to test the performance of our methods. One of the data sets is a generic text classification data set of Reuters news articles. The second data set is a collection of documents that has been labelled using TAR by one of the clients.

3.5.1 RCV1-v2

The Reuter Corpus Volume 1 (RCV1) data set consists of $804{,}414$ categorised English newspaper articles [19] made available by Reuters, ltd. for research purposes. RCV1-v2, the corrected version of the corpus, is one of the common standards used to evaluate the quality of automatic document classification. The corpus is labelled for 103 topic codes, where one document can be assigned to multiple topics. Each document belongs to at least one topic and at most 17 topic. The size of the topics span five orders of magnitude, ranging from 5 occurrences up to $381{,}327$.

The corpus does not include the full text, solely the TF-IDF feature matrix and target matrix. The full text license agreement states that a feature matrix may be distributed as long as the original data can not be reconstructed. Therefore, words are removed from a large stop list and the remaining words are stemmed. As a result, the TF-IDF matrix consists of $47{,}236$ features.

Statistics of the RCV1 corpus such as description of the topic, number of documents in the category and prevalence can be found in Table A.1 in the Appendix.

3.5.2 Client data

To test the performance of the methods on a real-world case, client data is used on which TAR was performed and the responsiveness of all documents is known. Due to confidentiality, we are unable to disclose which client or what kind of case it is. We received preprocessed TF-IDF vectors and labels for the documents. The case consists of $99{,}957$ documents and the TF-IDF matrix consists of $901{,}956$ features. Of these documents, $10{,}206$ are responsive to the issue.

Even though the feature space of the client data is significantly larger than the feature space of the Reuters data, it is possible that the Reuters data set is capable of simulating the TAR process with similar results as the client data.

3.6 Implementation

The full implementation of the experiments was done in Python. The linear SVM was implemented using version 2.11 of LIBLINEAR [6] which allows weights for data instances to passed during training. LIBLINEAR is an open source library that uses a coordinate descent method to solve the optimisation problem [13]. We adjusted the LIBLINEAR package to support probability outputs as described in the LIBLINEAR FAQ [1]. Both the SVM and BSVM are implemented using LIBLINEAR.

The Transductive SVM is implemented using SVMlin. This package implements a Linear Transductive L2-SVMs with multiple switchings [35]. We use version 1.0 of the SVMlin package. For the calculation of the NMF topic models we used the scikit-learn implementation using default parameters. Scikit-learn version $0.19.0$ [27] was used implement NMF. The number of components used to calculate NMF was 20.

[1] https://www.csie.ntu.edu.tw/ cjlin/liblinear/FAQ.html

3.7 System information

All the experiments were run on the DSLab servers from Leiden University. In particular, *Latinum*, *Adamant* and *Octiron*. All three servers are equipped with 16 Intel Xeon E5-2630v3 CPUs @ 2.40GHz (32 threads). Latinum has 1.5TB RAM and Adamant and *Octiron* 512GB RAM.

Chapter 4

Experimental setup

Multiple experiments were performed to research the possible reduction in manual labour. All the different experiments were conducted using the same experimental framework. Within this framework, the TAR process is simulated to measure the difference in performance between the baseline and the individual components. A large number of statistics were measured during the simulation and these measures were used to compare the performance. This way, the components best suitable to reduce the manual labour in TAR were identified. The same experimental framework was used to measure and validate the effect of our novel techniques Sampled labelling and Topic Model-Based Filtering.

First the experimental framework and the metrics which are stored during the simulation are described. Afterwards, the experiments performed within the experimental framework are specified.

The performance is compared by simulating the TAR process as if it were a real legal case. The first step is to create a seed set as a starting point for the simulation. The seed set consists of 500 random responsive and 500 random non-responsive documents. For the Reuters data set, all documents from categories that are not the responsive category make up the complete non-responsive set and 500 are randomly selected from these non-responsive categories for the initial seed set. For the client data, the labels define the responsive and non-responsive documents and 500 are selected at random to be the initial seed set. In legal review, the seed set is often constructed using search. The seed set is used to train a classifier and in turn the resulting model classifies all unlabelled documents. The probability output ranks the documents from most-probable responsive to least-probable responsive. When using the CAL protocol, a batch of the top 1,000 documents are presented to the reviewer and are manually labelled. Given that the data sets are fully annotated, the manual labour is simulated and the documents are labelled instantly.

The labelled documents are added to the training data of the next iteration and another classifier is trained which in turn classifies all unlabelled documents again. This iterative process continues until 90% recall has been achieved or 100,000 documents have been manually labelled. Within this TAR simulation, three experiments will be conducted. To ensure a valid comparison, the seed set is kept the same for all experiments.

4.1 Evaluation

To evaluate the performance of the classifiers, a number of evaluation measures are employed. The precision of a classifier indicates how many of the selected documents are in fact responsive. Formula 4.1 shows the precision, where tp is the number of true positives and fp is the number of false positives.

$$Precision = \frac{tp}{tp + fp} \tag{4.1}$$

The recall of a classifier measures how many of the total responsive documents are found. Formula 4.2 calculates the recall, where fn is the number of false negatives.

$$Recall = \frac{tp}{tp + fn} \tag{4.2}$$

For both these measures, a variant is used to measure the performance of the classifier for every iteration in the TAR process. Since the labels of the documents are unknown, the documents that are manually reviewed are used to calculate the *return set recision*. Given a batch of documents that the classifiers deems likely to be responsive, how many are of the documents are indeed responsive. The *Full corpus recall* calculates what percentage of responsive documents have been found by dividing the number of responsive documents in the current training set by the total number of responsive documents in the corpus. Therefore, the full corpus recall can only be calculated when all the labels are known. Nevertheless, the full corpus recall is very useful in research.

An important metric to measure the performance of the classifier in the TAR process is the F_1score. The F_1score is the harmonic mean of the recall and precision and is formulated as follows:

$$F_1score = 2 * \frac{precision * recall}{precision + recall} \tag{4.3}$$

The performance of the labour reduction methods within the simulation is evaluated based on various measures. These measures are used to analyse how the process is progressing. Every iteration of the simulation, the following results are stored:

- The amount of manual labour om terms of number of documents labelled
- Responsive documents found
- Non-responsive documents found
- Number of unlabelled documents left
- F1-score on unlabelled
- True positives, True negatives, False positives, False negatives
- Return set precision
- Full-corpus recall
- Time elapsed
- Parameters

Using these measures, the performance of the algorithms is compared in a number of ways. One very important measure used in the experiments is the *Manual labour for Recall*. This value indicates for each algorithm how many documents need to be reviewed to reach a given recall (e.g. 90%).

The goal of the first experiment is to measure the extend to which the topic modelling distances can be used to classify documents as responsive or non-responsive. The goal of the second experiment is to identify the best method to increase the return set precision. In the last experiment, the effect of sampled labelling on required manual labour is investigated and the precision of sampled labelling is researched.

4.2 Topic Model-Based Filtering

The Topic Model-Based Filtering experiment does not go through the complete TAR simulation and exclusively uses the seed set of the TAR simulation. The Topic modelling filtering component selects documents that are non-responsive with a high certainty. We use all available categories from the Reuters RCV1 corpus in the topic modelling experiment. Information about all Reuters categories such as description, number of documents and prevalence can be found in table A.1 in the Appendix.

Before the start of the TAR simulation, the NMF topic models for each document in the corpus are calculated. The seed set is used to calculate the centroid of the NMF topic vectors for the topic. One of parameter experimented with is the number of topics to calculate the NMF topics for. The distance from the centroid to all the instances is calculated and two distance distributions as described in Section 3.2.1 are created.

We experiment with Topic Model-Based Filtering results to determine the number of documents that can be filtered for each category and what the precision is of the documents that are filtered. In the experiment, we test our method of defining the number of documents to be filtered, we want to filter as many negative documents as possible without filtering many responsive documents. In order to quantify the performance of the topic modelling filtering, we measure the number of filtered documents, the number of documents in the filtered set that are responsive (false negatives) and the lost recall by filtering these responsive documents.

Two different settings for p, namely $p = 0.01$ and $p = 0.05$, have been compared to research the effect of this threshold on the precision of the filter. The number of bins used to bin the distances is set to 50. In the experiment focused on increasing the return set precision, Topic Model-Based Filtering is used as a method to increase the amount of training data and thereby increasing the return set precision.

4.3 Improving return set precision in the simulation

The second experiment within the TAR simulation focuses on increasing the return set precision. In chapter 3 we identify various methods to reduce the manual labour required to reach a certain recall by increasing the return set precision. The performance of these methods is compared to the performance of the baseline. The implementation of the methods are described as components that are added to the baseline. Components can either be adjustments or additions to the baseline algorithm. Most components are independent and can be combined.

The baseline algorithm is a linear SVM using the default parameter $C = 1$. Initial benchmarks show that the SVM outperforms k-NN and Rocchio in the topic classification task [18]. The results of the baseline will be indicated as *SVM*.

Topic Code	Description	Number of documents	Prevalence
GVIO	War, civil war	32,615	4.0%
GSPO	Sports	35,317	4.4%
C11	Strategy / Plans	24,325	3.0%
C13	Regulation / Policy	37,410	4.6%

Table 4.1: Characteristics of topic codes in RCV1 corpus used for experiments

From the Reuters RCV1 corpus, the topics chosen to evaluate the performance of algorithms are shown in Table 4.1. Previous work has shown that well-defined and clear categories such as "sports" achieve higher performance than more vaguely defined categories such as "strategy/plans" [36].

The five components which are compared to the baseline are described below. The parameters used in the experiment are elaborated on here.

1. **Self-training**
   The self-training components applies self-training to the classifier. Since this is an addition to the classifier, the self-training results will be indicated with the prefix *ST-*.

2. **BSVM**
   To increase the return set precision of the classifier, this component changes the classifier from an SVM to a BSVM. The BSVM implementation uses the unlabelled data as part of the training data and afterwards predicts the responsiveness of each of the unlabelled documents. The default parameters for the BSVM are used, meaning $C_p = C_n = 1$. Since the BSVM is an alternative classifier, the results are indicated as *BSVM*. The BSVM can be combined with components that are additions to the classifier.

3. **TSVM**
   The TSVM component changes the classifier from an SVM to a TSVM. The default parameters will be used, therefore $C_p = C_u = 10^{-5}$. The TSVM will be indicated in the results as *TSVM*. Same as the BSVM, the TSVM can be combined with other components whenever those components are additions to the classifier.

4. **Applied Topic model-based filtering**
   Topic Model-Based Filtering is done to create a set of filtered documents. These filtered documents are used to decrease the size of the unlabelled set and increase the size of the training data, which could yield an improvement in performance. Since this is an addition to the classifier, the results are shown with the prefix *TMF-* and the component can be combined with other components.

5. **Grid Search**
   The Grid Search component implements grid search parameter optimisation to find the optimal weights every iteration. For Grid Search it is important to research at which point the extra labour to create the hold-out validation set is saved by the increased performance. We have chosen a random validation set size of 1%. The validation set size will be rounded to the closest thousand so the manual labour can be expressed in thousands of documents reviewed, which allows for the comparison of performance with other components for every iteration in the TAR process. We investigate whether using Grid Search with a validation set of this size is capable of improving the return set precision. A larger validation set makes it significantly harder to reach the break-even point. For the Reuters data set, containing 804,414 documents, the validation set size is 8,000 documents.

   The weights optimised with the Grid search are C_p and C_n, for the positive and negative instances. For the regular SVM and the BSVM, the search space for C_p ranges from 1 to 0.01 in 20 steps and the search space for C_n ranges from 1 to 0.001 in 20 steps. The grid will therefore search 400 combinations of parameters. The experiment we did only includes Grid Search as addition to the baseline SVM but could be tested for other components in the future.

   The results of the grid search component are indicated using the prefix *GS-*. Grid search can be combined with different components.

4.4 Sampled labelling

This experiment focuses on Sampled Labelling. The effect of the precision threshold on the precision of the skipped documents is researched. Furthermore, we research which problems are suitable for the use of Sampled labelling. For the Sampled Labelling experiment, all available categories from the Reuters RCV1 corpus are used.

The labelling method in the TAR simulation is adjusted to Sampled Labelling and the full simulation is ran. In addition to all the performance measures for the TAR simulation, the number of skipped documents and the number of non-responsive documents that are automatically labelled are measured. These measures are used to calculate the precision of the skipped documents.

The settings for Sampled Labelling are as follows. The initial *skip size* is set to 400. This number was chosen since it can be easily divided by two and leave round numbers. The *batch size*, the number of documents to be reviewed to calculate the return set precision, is set to 100. The method needs two more parameters, the *skip threshold* indicates whether the documents should be skipped or not and the *double threshold* indicates whether the *skip size* should be doubled or not. When the return set precision of the reviewed batch is below the *skip threshold*, the *skip size* is divided by two. In the Sampled Labelling experiment, the *skip threshold* will be set to 0.9, 0.95 and 0.98 and the *double threshold* is set to 0.95, 0.97 and 0.99. The *double threshold* must always be larger than or equal to the *skip threshold*.

The simulation is terminated when sampled labelling has not been used for three iterations of the simulation in a row. Since the return set precision is highest at the start of the TAR simulation, if the return set precision is not high enough at the start of the simulation, no Sampled Labelling will be used. Hence, the labelling protocol is equal to the regular CAL protocol. The categories which will benefit from this new labelling protocol are identified in this experiment.

After experimenting with the parameters for all the Reuters categories, the Sampled Labelling experiment continues by researching the effect of Sampled Labelling on the manual effort needed to reach a certain recall. For this part of the experiment, the same four Reuters categories were used as in the *Increasing return set precision* experiment. Sampled Labelling was added as a component with the prefix *SL-*. This experiment was performed for all three sets of parameters. The component from the previous experiment that performs best was used as the basis on which Sampled Labelling is applied.

Chapter 5

Results

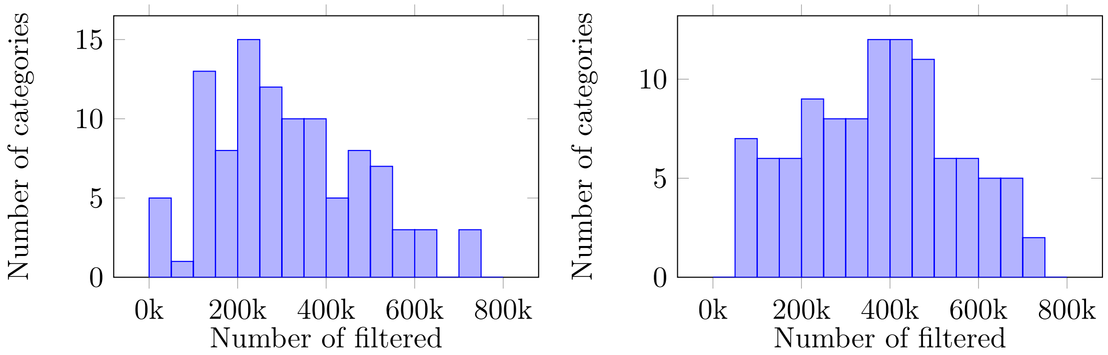
In this chapter, the results of our experiments are presented. First, the Topic model-based filtering experiment are presented, followed by the *Increase return set precision* results and finally the Sampled labelling results. The results for each experiment will be split into a section about experiments on the Reuters corpus and a section about the experiments on the client data.

5.1 Topic model-based filtering

In this section, we present the results of the Topic model-based filtering experiment. Starting with the results of the experiment on the Reuters data set, followed by the results using the client data.

Reuters

The full table with results of the Topic model-based filtering experiment for $p = 0.01$ and $p = 0.05$ are shown in the Appendix in Tables B.1 and B.2 respectively. To summarise the results, three pairs of histograms show the number of filtered documents, the number of filtered responsive documents and the lost recall by filtering these responsive documents for each of the 103 categories from Reuters. Figure 5.1 shows the number of categories binned for the number of documents that are filtered using topic model-based filtering. For $p = 0.01$, an average of $312{,}532$ documents are filtered for each category. An average of 40% of the unlabelled documents are filtered. Topic model-based filtering filters a maximum of $736{,}079$ documents for the category containing documents about Sports. For $p = 0.05$, an average of $372{,}494$ documents are filtered for each category, which translates to an average of 47% of the unlabelled documents. A maximum of $748{,}509$ documents are filtered, also in the category about Sports.

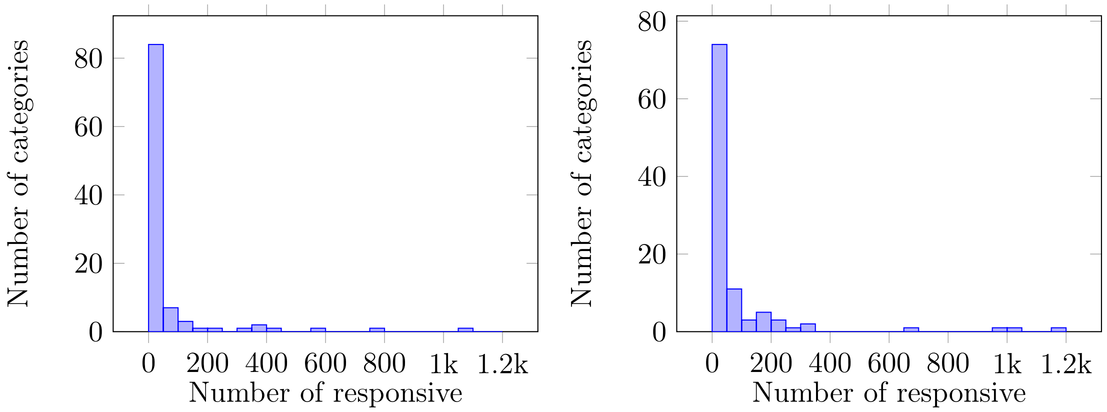


(a) $p = 0.01$

(b) $p = 0.05$

Figure 5.1: Histogram of number of filtered documents for each of the categories

The number of false negatives, filtered responsive documents, are shown in Figure 5.2. For $p = 0.01$, an average of 59 responsive documents are filtered. A maximum of $1{,}084$ responsive documents are skipped, the category where these responsive documents are skipped contains $204{,}820$ responsive documents in total. 22 out of 103 categories in the Reuters corpus filter no responsive documents and 53 out of 103 categories filter 10 or less documents. For $p = 0.05$, an average of 81 responsive documents are wrongly filtered. A maximum of $1{,}194$ documents are skipped for a category that contains $151{,}785$ responsive documents. 14 out of 103 categories filter no responsive documents and 43 out of 103 categories filter 10 documents or less.

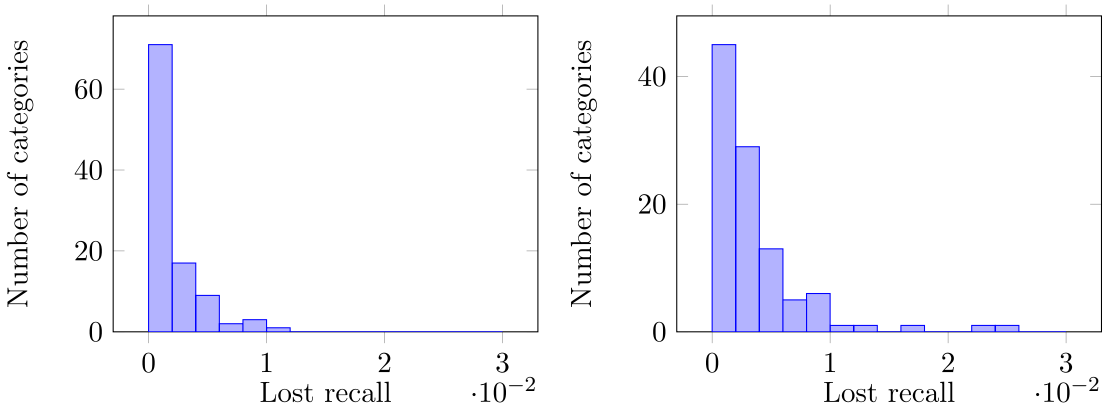


(a) $p = 0.01$

(b) $p = 0.05$

Figure 5.2: Histogram of number of filtered responsive documents for each of the categories

The last pair of histograms, shown in Figure 5.3, displays the lost recall. The lost recall is the decrease in maximum attainable recall in the TAR process after filtering the documents. This decrease is caused by responsive documents that are wrongly filtered. For $p = 0.01$, the lost recall is 0.18% on average. The maximum lost recall is 1.03%, meaning the maximum attainable recall in TAR will drop from 100% to 98.97%. 93 of the 103 categories have a lost recall below 0.5%. For $p = 0.05$, the average lost recall is 0.3% and the maximum lost recall is 2.4%. The number of categories with a lost recall below 0.5% is 80.



(a) $p = 0.01$

(b) $p = 0.05$

Figure 5.3: Histogram of lost recall for each of the categories.

Client data

The same experiment was done using client data. For $p = 0.01$, 22,169 documents are filtered. Of these documents, 8 documents are in fact responsive. Therefore, the lost recall for the experiment on the client data is 0.07%. Topic model-based filtering filters 25% of the unlabelled documents. For $p = 0.05$, 25,670 documents are filtered. Of these documents, 23 documents are in fact responsive, this means there is a lost recall for the experiment on the client data of 0.22%. 28% of the unlabelled documents are filtered in the experiment.

5.2 Improving return set precision in the simulation

For each of the categories that we use to simulate the TAR process, we have created a Gain Chart which plots the recall achieved at a certain point in the simulation against the manual reviewing labour performed up to that point.

A partial result for the category *Strategy / Plans* is shown in Figure 5.4. We can see that the SVM performs best in the beginning of the TAR simulation but is overtaken by the TSVM. The GS-SVM starts with 8,000 documents already labelled to create the validation set, but has a higher return set precision and therefore reaches higher recall than the SVM. The complete collection of gain charts can be found in Appendix C.

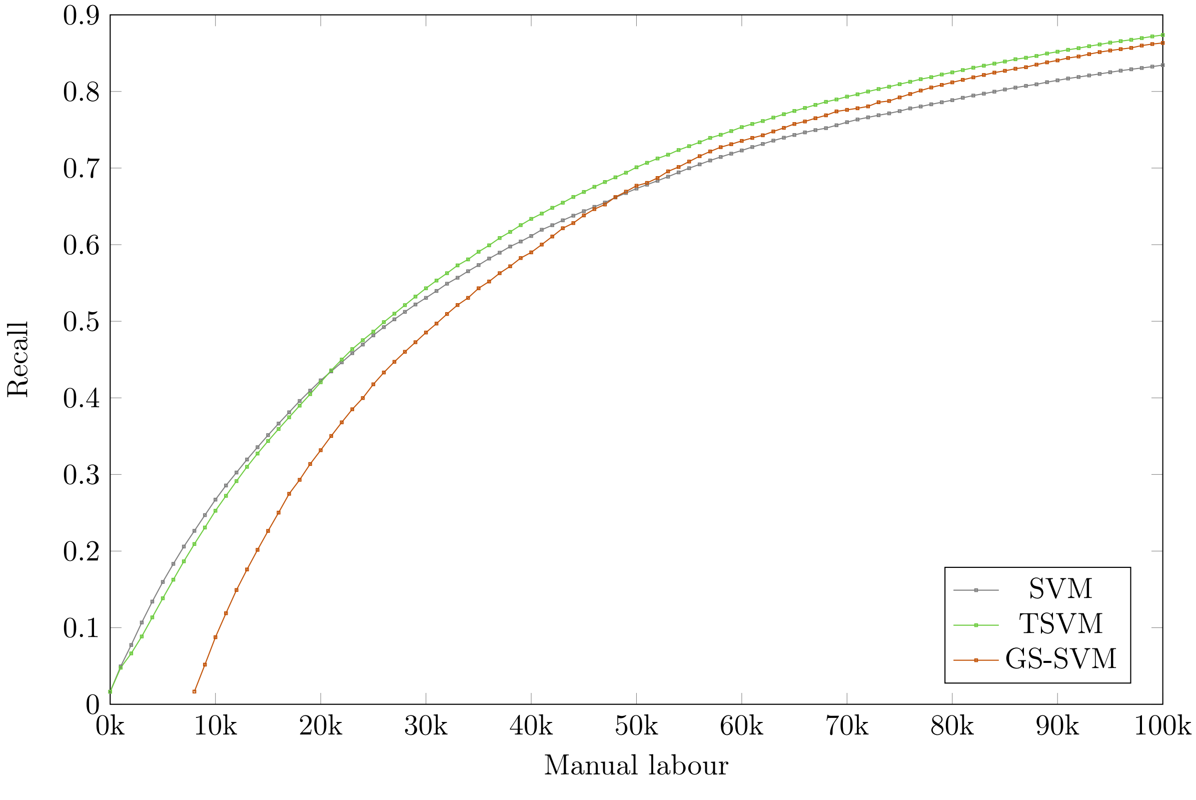


Figure 5.4: Gain Chart: Strategy / Plans

In order to show the progression of the return set precision, this value has been plotted for every iteration. Each iteration represents a batch of 1,000 manually reviewed documents. We have chosen to show the iterations instead of the manual labour on the x-axis in order to paint a clear picture of the return set precision at any point in the TAR process, especially for the Grid Search SVM that starts at 8,000 reviewed documents because of the validation set. Due to the high dimensionality of the training data for the client data, we were unable to run the BSVM experiment on the client data. The LIBLINEAR package had a bug that caused the implementation to fail when training with data containing more than 2^{32} zero values. Since the BSVM uses all the available data to train, an error occurs.

Table 5.1 presents the results for the Increase return set precision experiment. The top two categories, *Strategy / Plans* and *Regulation / Policy*, show the amount of manual labour necessary to reach 0.8 recall since 0.9 recall was not reached within 100,000 labelled documents. The bottom two categories, *Sports* and *War / Civil War*, show the amount of labour needed to reach 0.9 recall.

	Experiment	Baseline	ST-SVM	BSVM	TSVM	TMF-SVM	GS-SVM
Recall > 0.8	Strategy / Plans	85,000	81,000	85,000	**73,000**	82,000	77,000
Recall > 0.8	Regulation / Policy	87,000	84,000	87,000	**78,000**	83,000	85,000
Recall > 0.9	Sports	32,000	32,000	32,000	32,000	32,000	40,000
Recall > 0.9	War / Civil War	46,000	**44,000**	46,000	**44,000**	45,000	51,000
Recall > 0.9	Client data	24,000	25,000	-	**20,000**	24,000	23,000

Table 5.1: Minimum manual labelling effort needed to reach recall

Table 5.2 shows the amount of second it takes to execute an iteration for each of the components. The longest execution time is needed by the Grid Search SVM, which trains 400 SVM's to find the right parameters. The fastest component is the Topic Model-based filtered SVM.

	Average time (s)
SVM	95
ST-SVM	198
BSVM	100
TSVM	168
TMF-SVM	**71**
GS-SVM	611

Table 5.2: Average time per iteration of the TAR process

5.3 Sampled labelling

The full results of the Sampled Labelling experiment can be found in the appendix in Tables D.1, D.1 and D.1. A table for each of the three different settings of the parameters. Below, the results are consolidated by comparing the number of categories where Sampled Labelling had an effect, the average number of documents skipped, the average number of false positives and the average gained recall. The gained recall is the amount of recall that was gained by skipped documents that were not manually labelled. The table also shows the maximum skipped number of documents, the maximum gained recall of any category and the minimum precision of any category. In Table ?? below, t_s is the skip threshold and t_d is the double threshold.

	$t_s = 0.9$ $t_d = 0.95$	$t_s = 0.95$ $t_d = 0.97$	$t_s = 0.98$ $t_d = 0.99$
Number of categories	63	49	43
Average skipped documents	21,956	24,467	21,667
Average false positives	820	610	287
Average gained recall	0.336	0.326	0.249
Average precision	0.942	0.946	0.979
Maximum skipped documents	305,400	283,400	237,700
Maximum gained recall	0.923	0.883	0.847
Minimum precision	0.870	0.925	0.944

Table 5.3: Sampled labelling results

Table 5.4 shows the results of the experiments of the four Reuters categories with Sampled Labelling enabled. Since the TSVM displayed superior performance in the previous experiment, this was the base implementation on top of which Sampled Labelling is applied. The results for all three sets of parameters are shown below.

	Category	Experiment Measure	TSVM	SL-TSVM [1]	SL-TSVM [2]	SL-TSVM [3]
Recall >0.8	Strategy / Plans	Labour	73,000	73,000	73,000	73,000
		Skipped	-	0	0	0
		FP	-	0	0	0
Recall >0.8	Regulation / Policy	Labour	78,000	78,000	78,000	78,000
		Skipped	-	0	0	0
		FP	-	0	0	0
Recall >0.9	Sports	Labour	32,000	1,000	2,000	3,000
		Skipped	-	32,800	32,100	30,200
		FP	-	517	408	243
Recall >0.9	War / Civil War	Labour	44,000	33,000	40,000	42,000
		Skipped	-	9,200	3,400	1,300
		FP	-	775	186	57

Table 5.4: Sampled labelling experiment on Reuters categories

[1] $t_s = 0.90; t_d = 0.95$

[2] $t_s = 0.95; t_d = 0.97$

[3] $t_s = 0.98; t_d = 0.99$

Chapter 6

Discussion

This chapter is where we discuss the results and the implications of the results. We will start by discussing the results of the Topic model-based filtering experiment, followed by the Improving the return set precision in the simulation experiment and finally the Sampled Labelling experiment.

6.1 Topic model-based filtering

The value of p can be used to set the threshold of how many documents should be filtered and to what degree of certainty the filtered documents should not contain responsive documents. A lower value for p will result in a lower average lost recall, meaning less filtered responsive documents. A higher value for p will result in more documents being filtered.

The results show that a relative large number of documents can be filtered, over 40% for both parameters, while maintaining a high precision. The maximum lost recall is close to 1% for $p = 0.01$, which can be argued by lawyers to be acceptable since the goal of TAR is often to reach a recall well below 100%.

Topic model-based filtering can be applied to filter noise and significantly increase the number of negative training documents before even starting the TAR simulation. This decrease in unlabelled documents leads to a decrease in time needed per iteration which could allow for more computationally expensive methods to be performed within the same time frame as a regular SVM. The noise filtering could also be used to decrease the noise in the TF-IDF matrix by recalculating it after the documents have been filtered. Future work could research whether this has an impact on the overall TAR performance.

The experiments show that using Topic model distances to the centroid are an effective way to filter non-responsive documents.

6.2 Improving return set precision in the simulation

The four Reuters categories can be divided in two categories, easy and hard. *Sports* and *War / Civil War* are categorised as easy TAR tasks, *Strategy / Plans* and *Regulation / Policy* are categorised as hard tasks. As seen in Figure C.8, the return set precision is near 1 during a significant part of the simulation. Therefore, *Sports* is a category that does not leave a lot of room for improvement of the return set precision. None of the proposed components for increasing the return set precision shows a decrease in manual labelling effort needed to reach the required recall.

The Self-training SVM (ST-SVM) shows a decrease in labelling effort necessary to reach the required recall in three out of four categories. Interestingly, judging from Figures D.1 and C.2, the ST-SVM outperforms all other components in the first half of the simulation but is outperformed in the second half.

The results of the Biased SVM (BSVM) are identical to the baseline. No significant improvement is found when using the default parameters. Our hypothesis is that the performance of the BSVM is highly dependent on choosing the right parameters.

The component with the most significant decrease in manual labelling effort needed to reach the required recall is the Transductive SVM (TSVM). For each category it requires the lowest amount of labelling effort needed. The TSVM shows superior performance compared to all other proposed components. During the first half of the TAR simulation, the ST-SVM does have a higher return set precision, but the TSVM has higher return set precision in the second half of the simulation. Averaged over all four categories, the TSVM decrease the amount of labelling necessary by 7%. The most significant decrease in labelling effort is for the category *Strategy / Plans* with a decrease of 14.1%.

The Applied Topic Model Based filtering component (TMF-SVM) shows a small increase in return set precision and therefore decrease in necessary manual effort. Noteworthy is that the average time per iteration decreases when using this component. This is caused by the fact that not training the SVM but predicting the labels is the most time-consuming task each iteration. By significantly increasing the amount of (negative) training data by filtering up front, less labels have to be predicted and the process is therefore sped up.

The Grid Search SVM (GS-SVM) starts behind the other approaches due to the fact that the validation set has to be labelled for this technique. For the categories *Sports* and *War / Civil War*, the increase in return set precision is not big enough to make up for this. However, for the categories *Strategy / Plans* and *Regulation / Policy*, the increase in return set precision is significant enough to reach the break-even point between regular SVM and GS-SVM before 0.8 recall, therefore reaching 0.8 recall with less manual reviewing effort.

The TSVM performs best for the client data experiment as well, reducing the amount of manual labour from $24{,}000$ reviewed documents to reach 90% recall to 20.000 documents, a decrease of 16.7%. The financial implications of reducing the manual labour are very significant. If lawyers would have to review 16.7% less documents, this could save an equal percentage in costs for the complete TAR process. Not all the components show similar results as the TSVM on the client data. The ST-SVM even performs slightly worse than the baseline implementation. The TMF-SVM shows no improvement aside from reducing the time per iteration and the GS-SVM shows a modest reduction in labelling effort.

Overall, increasing the return set precision is an effective method for reducing manual labour for harder TAR tasks. Easier TAR tasks leave less opportunities for enhancement and therefore the performance improvements are less significant using these techniques.

When comparing the Reuters experiments to the client data experiments, it stands out that the component that performs best is the same in both experiments. The *TMF-SVM* shows no difference in performance on the client data, while for the experiments with the Reuters data it does cause a reduction in necessary labour. The *GS-SVM* only performs better for the harder tasks. It could be argued that, given the return set precision graph of the client data, the experiment with the client data falls in this more difficult category. Hence, the results of the *GS-SVM* are also similar for the client data and Reuters data. The *ST-SVM* even causes a small increase in necessary labelling effort for the client data, while for the Reuters data the labelling effort decreases. Even though the results of the client data are not completely similar to the results of the Reuters data, and a Reuters category could maybe be found that is more similar to the client data, the results on the Reuters data could give a decent indication of the performance on the client data. If more client data is available to experiment with, better research can be done into the effectiveness of using Reuters data to simulate client data experiments.

6.3 Sampled Labelling

In order for Sampled Labelling to skip documents, the precision of the sampled batch must be above the threshold. Therefore, the number of categories on which Sampled Labelling has an effect changes depending on the threshold. Using the highest threshold, 43 out of 103 categories in the Reuters corpus use Sampled labelling. Likewise, the average precision of the skipped documents changes between thresholds, with a higher thresholds corresponding to higher precisions. On average, over $20{,}000$ documents are skipped for each of the settings and can therefore reduce the amount of labour necessary significantly. When $20{,}000$ documents are skipped, more than 20 review iterations of $1{,}000$ document' batches are saved.

From the results of the previous experiment, in Figures C.6 and C.7, we know that the experiments *Strategy / Plans* and *Regulation / Policy* never reach a return set precision above 90%. Therefore, Sampled Labelling has no effect on these categories and performs exactly the same as the regular CAL protocol.

For the two other categories, which achieve higher return set precision in the previous experiment, Sampled Labelling has a significant impact. The amount of labour required to reach the required recall is decreased considerably. For the category *Sports* the decrease in labour ranges from 90.6% to 96.9%, reaching 90% recall after only labelling $1{,}000$ documents for $t_s = 0.90$ and $t_d = 0.95$. For the *War / Civil War* category, the reduction is smaller but still very significant. The decrease in labour ranges from 4.5% to 25.0%, depending on the threshold.

The return set precision of this particular client data is not high enough for Sampled Labelling to have an effect. Future work could research the effect on a client data set that does reach the required precision threshold. On the Reuters data set, Sampled Labelling proves to be an effective method for reducing the amount of manual labour.

Chapter 7

Conclusion

In this chapter, we share our conclusions and answer the research questions. After answering the research questions, we describe what possible future work could lead to valuable research insights.

7.1 Research Questions

1. *How can the necessary amount of manual labour during legal review be reduced?*

   We have identified three methods for reducing the amount of manual labour necessary to reach the required recall: (1) reducing the size of the problem, (2) increasing the return set precision and (3) changing the protocol by which the documents are labelled. For each of the three methods we set up experiments to research the effect on performance. Due to the strict protocols, options for changing the way documents are labelled are limited. Sampled Labelling is an addition to the existing CAL protocol.

2. *What is the effect of the identified methods to reduce the amount of manual labour?*

   The results of our experiments show the effect of all identified methods for reducing the amount of manual labour. For increasing the return set precision, the *TSVM* shows the most significant labour reduction, especially in more difficult tasks. For tasks that start with a high return set precision, we introduced a novel labelling technique, Sampled Labelling, which shows very promising results. In our experiments Sampled Labelling considerably reduced the amount of necessary labour. By combining these two techniques, our research found methods to improve the performance of both easy and hard tasks, potentially leading to a significant cost reduction of fulfilling a request for production.

   Defensibility of the methods leading to these potential reductions in manual labour is important. Applying an improved classifier like TSVM is very defensible since it still works within the same TAR protocol, Continuous Active Learning. However, Sampled Labelling does change the TAR protocol and defensibility could be affected. Given the fact that, using Sampled Labelling, no documents can be automatically labelled as non-responsive, we believe that defensibility will not be a big issue.

3. *To what extent does the performance of the TAR simulation using the Reuters RCV1 corpus reflect the performance of the TAR simulation using client data?*

   From the experiments we did with one client data set, we are unable to generalise how good the performance of the TAR simulation using Reuters data reflects the performance using client data. However, for this client data set, the classifier that improved the baseline the most is the same for both corpora and it seems that an improved performance on the Reuters corpus also indicates an improvement on this particular client data set.

7.2 Future work

A lot of research can still be done to improve the performance of the TAR simulation. In this section we will present some ideas for future research.

Topic Model-Based Filtering can be used before starting TAR to filter noise. Especially the client data includes a lot of noise, which is apparent from the almost one million terms in the TF-IDF matrix. Future work could try to figure out, whether using the filtering method and afterwards recalculating TF-IDF with reduced noise could improve performance. Additional research could be done to investigate what the effect is on performance when combining Grid Search with different components such as TSVM and BSVM. Our hypothesis is that the BSVM relies heavily on the parameters to perform well. An experiment with GS-BSVM compared to GS-SVM could be used to proof this hypothesis.

Sampled Labelling has not yet been tested on client data that does reach a high enough return set precision. Future work could research the effect of Sampled Labelling on client data by getting data of a completed TAR process that does reach the threshold.

Future research could look into what specific Reuters categories are best suitable for simulating the TAR process. In order to do so, more client data would be necessary to provide a better picture of the scope of the differences from corpus to corpus and issue to issue. With the limited client data we have performed experiments with, we were unable to research which categories are best suitable for simulating TAR and how good the reflection of performance is for client data in general.

Shifting from supervised to semi-supervised learning is a step in the right direction of improving the TAR performance. However, from the results we can learn that some classifiers work better in different segments of the TAR process. A Reinforcement Learning approach might be able to learn from the current state of the TAR process, what classifier to use next. Since SAL with the use of uncertainty sampling can be used to increase the performance of the classifier by selecting and labelling documents that the classifier is least certain about. Research could look in whether switching between TAR protocols could be an effective method to improve the performance and reduce manual labour even further.

Bibliography

[1] Charu C Aggarwal and Chengxiang Zhai. A SURVEY OF TEXT CLUSTERING ALGORITHMS.

[2] Thomas I Barnett and Svetlana Godjevac. Faster, better, cheaper legal document review, pipe dream or reality? pages 1–16.

[3] Jianlin Cheng, Amanda Jones, Caroline Privault, and Jean-michel Renders. Soft Labeling for Multi-Pass Document Review. *pdfs.semanticscholar.org*, pages 1–11, 2011.

[4] Gordon V. Cormack and Maura F. Grossman. Waterloo (Cormack) participation in the TREC 2015 Total Recall Track. *The Twenty-Fourth Text REtrieval Conference Proceedings (TREC 2015) , NIST Special Publication 500-319*, page 3, 2015.

[5] Gordon V. Cormack and Maura R. Grossman. Evaluation of machine-learning protocols for technology-assisted review in electronic discovery. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval - SIGIR '14*, pages 153–162, 2014.

[6] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[7] Cédric Févotte and Jérôme Idier. Algorithms for Nonnegative Matrix Factorization with the β-Divergence. *Neural Computation*, 23(9):2421–2456, 2011.

[8] Maura R Grossman and Gordon V Cormack. Inconsistent Assessment of Responsiveness in E-Discovery: Difference of Opinion or Human Error? 26(4):1–11, 2011.

[9] Maura R. Grossman and Gordon V. Cormack. The Grossman-Cormack glossary of technology-assisted review with foreword by John M. Facciola, U.S. Magistrate Judge. *Federal Courts Law Review*, 2013.

[10] Maura R Grossman, Gordon V Cormack, and Adam Roegiest. TREC 2016 Total Recall Track Overview. 2016(September):1–17, 2016.

[11] MR Grossman and GV Cormack. Technology-assisted review in e-discovery can be more effective and more efficient than exhaustive manual review. *Rich. JL & Tech.*, 2010.

[12] A N Hermans. Topic Model Selection for Forensic Cyber-crime Investigators: Defining and Finding an Optimal Number of Topics. 2017.

[13] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S Sathiya Keerthi, and S Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *Proceedings of the 25th international conference on Machine learning - ICML '08*, pages 408–415, 2008.

[14] T Joachims. Transductive Inference for Text Classification using Support Vector Machines. *ICML*, 99:200–209, 1999.

[15] Thorsten Joachims. Text Categorization with Support V ector Machines Learning with Many Relevant Features. *Machine Learning*, 1398(LS-8 Report 23):137–142, 1998.

[16] Amanda Jones, Marzieh Bazrafshan, Fernando Delgado, Tania Lihatsh, and Tamara Schuyler. The Role of Metadata in Machine Learning for Technology Assisted Review.

[17] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[18] David D Lewis and William A Gale. A Sequential Algorithm for Training Text Classiiers. 94:3–12.

[19] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[20] Ling Zhen, Naiyang Deng, Chunhua Zhang, and Junyan Tan. A new support vector machine for the classification of positive and unlabeled examples. In *11th International Symposium on Operations Research and its Applications in Engineering, Technology and Management 2013 (ISORA 2013)*, pages 169–176. Institution of Engineering and Technology, 2013.

[21] Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S Yu. Building Text Classifiers Using Positive and Unlabeled Examples.

[22] Larry M Manevitz, Malik Yousef, Nello Cristianini, John Shawe-Taylor, and Bob Williamson. One-Class SVMs for Document Classification. *Journal of Machine Learning Research*, 2:139–154, 2001.

[23] Mia Mazza, Emmalena K Quesada, and Ashley L Sternberg. In Pursuit Of Frcp 1: Creative Approaches To Cutting And Shifting The Costs Of Discovery IN PURSUIT OF FRCP 1: CREATIVE APPROACHES TO CUTTING AND SHIFTING THE COSTS OF DISCOVERY OF ELECTRONICALLY STORED INFORMATION. *Richmond Journal of Law and Technology Richmond Journal of Law & Technology*, 13(3), 2007.

[24] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. Foundations of Machine Learning. *Journal of Chemical Information and Modeling*, 53(9):1689–1699, 2013.

[25] Chapelle Olivier, Bernhard Schölkopf, and Alexander Zien. Semi-Supervised Learning. *Interdisciplinary sciences computational life sciences*, 1(2):524, 2006.

[26] Christopher H Paskach, F Eli Nelson, and Matthew Schwab. The Case for Technology Assisted Review and Statistical Sampling in Discovery. 2015.

[27] Fabian Pedregosa FABIANPEDREGOSA, Normalesuporg Alexandre Gramfort, Vincent Michel, Bertrand Thirion BERTRANDTHIRION, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer PETERPRETTENHOFER, Ron Weiss, Vincent Dubourg, Jake Vanderplas VANDERPLAS, Alexandre Passos, David Cournapeau, Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Bertrand Thirion, Peter Prettenhofer, Jake Vanderplas, Matthieu Brucher, Matthieu Perrot an Edouard Duchesnay PEDREGOSA, Al Matthieu Brucher MATTHIEUBRUCHER, Matthieu Perrot MATTHIEUPERROT, and Cea F Edouard Duchesnay EDOUARDDUCHESNAY. Scikit-learn: Machine Learning in Python Gaël Varoquaux. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[28] Stephen Robertson. Understanding Inverse Document Frequency: On theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520.

[29] HL Roitblat, A Kershaw, P Oot Journal of the Association for, and undefined 2010. Document categorization in legal electronic discovery: computer classification vs. manual review. *Wiley Online Library*.

[30] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[31] Johannes C Scholtes, Tim Van Cann, and Mary Mack. The Impact of Incorrect Training Sets and Rolling Collections on Technology-Assisted Review.

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

[33] Farial Shahnaz, Michael W. Berry, V. Paul Pauca, and Robert J. Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing and Management*, 42(2):373–386, 2006.

[34] Vikas Sindhwani and S Sathiya Keerthi. Large Scale Semi-supervised Linear SVMs.

[35] Vikas Sindhwani and S Sathiya Keerthi. Newton Methods for Fast Solution of Semi- supervised Linear SVMs.

[36] Jeroen Smeets. Boosting Search Recall by using Machine Learning and Automatic Document Classification. 2016.

[37] Michael Sperling, Rong Jin, Illya Rayvych, Jianghong Li, and Jinfeng Yi. Similar Document Detection and Electronic Discovery: So Many Documents, So Little Time. *Umiacs.Umd.Edu*, 2012.

[38] Paul P. Tallon. Corporate governance of big data: Perspectives on value, risk, and cost. *Computer*, 46(6):32–38, 6 2013.

[39] Isaac Triguero, Salvador García, and Francisco Herrera. Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. *Knowledge and Information Systems*, 42(2):245–284, 2015.

[40] David Van Dijk, Zhaochun Ren, Evangelos Kanoulas, and Maarten De Rijke. The University of Amsterdam (ILPS) at TREC 2015 Total Recall Track.

[41] V N Vapnik. The Nature of Statistical Learning Theory, 1995.

[42] Ellen M Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing and Management*, 36(5):697–716, 2000.

[43] Eugene Yang, David Grossman, Ophir Frieder, and Roman Yurchak. Effectiveness Results for Popular e-Discovery Algorithms. In *Proceedings of the 16th international conference on Artificial intelligence and law*, volume 417, pages 261–264, 2017.

Appendices

Appendix A

Reuters statistics

This Appendix contains a table which displays all the statistics about the Reuters Corpus. For each of the 103 Reuters categories, the category name, the category description, the number of documents in the category (#) and the prevalence (%) is shown. The largest category is *Corporate / Industrial* with $381{,}327$ documents and a prevalence of 47.4%. The smallest category is *Millennium issues* containing only 5 documents.

Cat	Description	#	%
C11	Strategy / Plans	24325	3
C12	Legal / Judicial	11944	1.5
C13	Regulation / Policy	37410	4.7
C14	Share listings	7410	0.9
C15	Performance	151785	18.9
C151	Accounts / Earnings	81890	10.2
C1511	Annual results	23211	2.9
C152	Comment / Forecasts	73092	9.1
C16	Insolvency / Liquidity	1920	0.2
C17	Funding / Capital	42155	5.2
C171	Share capital	18313	2.3
C172	Bonds / Debt issues	11487	1.4
C173	Loans / Credits	2636	0.3
C174	Credit ratings	5871	0.7
C18	Ownership changes	52817	6.6
C181	Mergers / Acquisitions	43374	5.4
C182	Asset transfers	4671	0.6
C183	Privatisations	7406	0.9
C21	Production / Services	25403	3.2
C22	New products / Services	6119	0.8
C23	Research / Development	2625	0.3
C24	Capacity / Facilities	32153	4
C31	Markets / Marketing	40509	5
C311	Domestic markets	4299	0.5
C312	External markets	6648	0.8
C313	Market share	1115	0.1
C32	Advertising / Promotion	2084	0.3
C33	Contracts / Orders	15332	1.9
C331	Defence contracts	1210	0.2
C34	Monopolies / Competition	4835	0.6
C41	Management	11355	1.4
C411	Management moves	10272	1.3
C42	Labour	11878	1.5
CCAT	Corporate / Industrial	381327	47.4
E11	Economic performance	8568	1.1
E12	Monetary / Economic	27100	3.4
E121	Money supply	2182	0.3
E13	Inflation / Prices	6603	0.8
E131	Consumer prices	5659	0.7
E132	Wholesale prices	939	0.1
E14	Consumer finance	2177	0.3
E141	Personal income	376	0.05
E142	Consumer credit	200	0.03
E143	Retail sales	1206	0.1
E21	Government finance	43130	5.4
E211	Expenditure / Revenue	15768	2
E212	Government borrowing	27405	3.4
E31	Output / Capacity	2415	0.3
E311	Industrial production	1701	0.2
E312	Capacity utilization	52	0
E313	Inventories	111	0
E41	Employment / Labour	17035	2.1
E411	Unemployment	2136	0.3
E51	Trade / Reserves	21280	2.6
E511	Balance of payments	2933	0.4
E512	Merchandise trade	12634	1.6
E513	Reserves	2290	0.3
E61	Housing starts	391	0.05
E71	Leading indicators	5268	0.7
ECAT	Economics	119920	14.9
G15	European community	20672	2.6
G151	Ec internal market	3307	0.4
G152	Ec corporate policy	2107	0.3
G153	Ec agriculture policy	2360	0.3
G154	Ec monetary / Economic	8404	1
G155	Ec institutions	2124	0.3
G156	Ec environment issues	260	0.03
G157	Ec competition / Subsidy	2036	0.3
G158	Ec external relations	4300	0.5
G159	Ec general	40	0.01
GCAT	Government / Social	239267	29.7
GCRIM	Crime, law enforcement	32219	4
GDEF	Defence	8842	1.1
GDIP	International relations	37739	4.7
GDIS	Disasters and accidents	8657	1.1
GENT	Arts, culture, entertainment	3801	0.5
GENV	Environment / natural world	6261	0.8
GFAS	Fashion	313	0.04
GHEA	Health	6030	0.7
GJOB	Labour issues	17241	2.1
GMIL	Millennium issues	5	0.001
GOBIT	Obituaries	844	0.1
GODD	Human interest	2802	0.3
GPOL	Domestic politics	56878	7.1
GPRO	Biographies / personalities	5498	0.7
GREL	Religion	2849	0.4
GSCI	Science and technology	2410	0.3
GSPO	Sports	35317	4.4
GTOUR	Travel and tourism	680	0.1
GVIO	War, civil war	32615	4.1
GVOTE	Elections	11532	1.4
GWEA	Weather	3878	0.5
GWELF	Welfare, social services	1869	0.2
M11	Equity markets	48696	6.1
M12	Bond markets	26036	3.2
M13	Money markets	53634	6.7
M131	Interbank markets	28185	3.5
M132	Forex markets	26752	3.3
M14	Commodity markets	85440	10.6
M141	Soft commodities	47708	5.9
M142	Metals trading	12130	1.5
M143	Energy markets	21957	2.7
MCAT	Markets	204820	25.5

Table A.1: Reuters categories with the number of documents in the category (#) and the prevalence (%)

Appendix B

Topic model based filtering results

This Appendix contains all the results of the Topic Model-Based Filtering experiments. The next two pages display two tables that show the number of filtered documents, False negatives and lost recall for each of the 103 Reuters categories. Table B.1 shows the results for $p = 0.01$ and Table B.2 for $p = 0.05$.

For $p = 0.01$, an average of $312{,}532$ documents are filtered for each category. An average of 40% of the unlabelled documents are filtered. Topic model-based filtering filters a maximum of $736{,}079$ documents for the category containing documents about Sports. For $p = 0.05$, an average of $372{,}494$ documents are filtered for each category, which translates to an average of 47% of the unlabelled documents. A maximum of $748{,}509$ documents are filtered, also in the category about Sports.

Cat	# Filtered	False N	Lost recall
C11	47768	0	0.0000
C12	17584	0	0.0000
C13	224871	120	0.0032
C14	100539	0	0.0000
C15	259662	174	0.0011
C151	376970	140	0.0017
C1511	371677	14	0.0006
C152	284898	332	0.0045
C16	107870	1	0.0005
C17	287047	379	0.0090
C171	220177	3	0.0002
C172	323712	15	0.0013
C173	405335	10	0.0038
C174	373048	4	0.0007
C18	136007	38	0.0007
C181	364083	51	0.0012
C182	343630	7	0.0015
C183	216568	4	0.0005
C21	147116	25	0.0010
C22	306088	31	0.0051
C23	350685	18	0.0069
C24	196757	55	0.0017
C31	101805	30	0.0007
C311	216259	0	0.0000
C312	298168	9	0.0014
C313	155228	2	0.0018
C32	195654	3	0.0014
C33	144658	11	0.0007
C331	265254	1	0.0008
C34	282331	2	0.0004
C41	246344	16	0.0014
C411	305334	62	0.0060
C42	106819	10	0.0008
CCAT	43949	799	0.0021
E11	143368	1	0.0001
E12	212593	26	0.0010
E121	50049	0	0.0000
E13	207500	3	0.0005
E131	325450	6	0.0011
E132	711414	0	0.0000
E14	115891	0	0.0000
E141	322362	0	0.0000
E142	491464	0	0.0000
E143	531206	3	0.0025
E21	3408	0	0.0000
E211	237396	26	0.0016
E212	16339	1	0.0000
E31	546656	12	0.0050
E311	474916	4	0.0024
E312	428719	0	0.0000
E313	324987	0	0.0000
E41	255433	61	0.0036
E411	482663	7	0.0033
E51	166607	35	0.0016
E511	222716	1	0.0003
E512	213620	30	0.0024
E513	501676	5	0.0022
E61	237570	0	0.0000
E71	730658	28	0.0053
ECAT	113220	417	0.0035
G15	399948	43	0.0021
G151	632805	9	0.0027
G152	186468	0	0.0000
G153	228140	1	0.0004
G154	394981	11	0.0013
G155	563926	10	0.0047
G156	504722	0	0.0000
G157	527635	3	0.0015
G158	475936	15	0.0035
G159	616679	0	0.0000
GCAT	198147	353	0.0015
GCRIM	302948	39	0.0012
GDEF	391425	13	0.0015
GDIP	297730	22	0.0006
GDIS	423500	78	0.0090
GENT	151481	2	0.0005
GENV	172479	5	0.0008
GFAS	110820	0	0.0000
GHEA	298880	9	0.0015
GJOB	228845	41	0.0024
GMIL	610833	0	0.0000
GOBIT	303876	0	0.0000
GODD	458164	11	0.0039
GPOL	511063	584	0.0103
GPRO	377910	4	0.0007
GREL	591915	26	0.0091
GSCI	284471	0	0.0000
GSPO	736079	49	0.0014
GTOUR	122556	0	0.0000
GVIO	429783	44	0.0013
GVOTE	464998	49	0.0042
GWEA	212891	10	0.0026
GWELF	216410	0	0.0000
M11	131574	4	0.0001
M12	499653	149	0.0057
M13	519455	226	0.0042
M131	478908	22	0.0008
M132	405243	24	0.0009
M14	354670	70	0.0008
M141	331103	19	0.0004
M142	287510	11	0.0009
M143	598112	56	0.0026
MCAT	264339	1084	0.0053

Table B.1: Topic model-based filtering results on the Reuters corpus for $p = 0.01$

Cat	# Filtered	False N	Lost recall
C11	429167	231	0.00950
C12	362826	86	0.00720
C13	144319	20	0.00053
C14	302767	5	0.00067
C15	324156	1194	0.00787
C151	380970	183	0.00223
C1511	610379	206	0.00888
C152	260961	250	0.00342
C16	90800	0	0.00000
C17	250320	184	0.00436
C171	210194	2	0.00011
C172	494806	66	0.00575
C173	382663	8	0.00303
C174	591028	57	0.00971
C18	384901	154	0.00292
C181	433850	131	0.00302
C182	525676	116	0.02483
C183	233857	6	0.00081
C21	133844	39	0.00154
C22	298585	30	0.00490
C23	124533	0	0.00000
C24	145360	20	0.00062
C31	67402	13	0.00032
C311	253540	5	0.00116
C312	333080	26	0.00391
C313	286355	6	0.00538
C32	370151	35	0.01679
C33	107799	8	0.00052
C331	327188	3	0.00248
C34	371702	33	0.00683
C41	238941	23	0.00203
C411	150597	2	0.00019
C42	82134	2	0.00017
CCAT	50240	977	0.00256
E11	419740	50	0.00584
E12	220995	32	0.00118
E121	411901	7	0.00321
E13	626243	57	0.00863
E131	540265	26	0.00459
E132	742791	5	0.00532
E14	473173	14	0.00643
E141	410848	0	0.00000
E142	491477	0	0.00000
E143	656922	7	0.00580
E21	462337	1031	0.02390
E211	96623	0	0.00000
E212	471928	311	0.01135
E31	553582	11	0.00455
E311	692244	16	0.00941
E312	428728	0	0.00000
E313	324970	0	0.00000
E41	214108	34	0.00200
E411	543850	12	0.00562
E51	158588	25	0.00117
E511	305286	26	0.00886
E512	154453	9	0.00071
E513	492272	5	0.00218
E61	237567	0	0.00000
E71	682265	19	0.00361
ECAT	55465	8	0.00007
G15	367118	35	0.00169
G151	670736	11	0.00333
G152	333628	5	0.00237
G153	476159	9	0.00381
G154	680960	35	0.00416
G155	570458	7	0.00330
G156	504715	0	0.00000
G157	576840	10	0.00491
G158	446814	8	0.00186
G159	616689	0	0.00000
GCAT	184119	300	0.00125
GCRIM	227552	12	0.00037
GDEF	375976	10	0.00113
GDIP	447347	80	0.00212
GDIS	202278	3	0.00035
GENT	256592	8	0.00210
GENV	204107	17	0.00272
GFAS	110821	0	0.00000
GHEA	381689	33	0.00547
GJOB	166069	15	0.00087
GMIL	610829	0	0.00000
GOBIT	301168	0	0.00000
GODD	395174	4	0.00143
GPOL	388840	60	0.00105
GPRO	403755	6	0.00109
GREL	454150	3	0.00105
GSCI	424387	15	0.00622
GSPO	748509	97	0.00275
GTOUR	99540	0	0.00000
GVIO	460220	60	0.00184
GVOTE	452968	35	0.00304
GWEA	252469	5	0.00129
GWELF	282692	4	0.00214
M11	552609	652	0.01339
M12	445390	78	0.00300
M13	502471	167	0.00311
M131	524288	38	0.00135
M132	465095	67	0.00250
M14	445699	247	0.00289
M141	613758	146	0.00306
M142	372552	29	0.00239
M143	558611	34	0.00155
MCAT	187346	184	0.00090

Table B.2: Topic model-based filtering results on the Reuters corpus for $p = 0.05$

Appendix C

Increase return set precision results

In this Appendix, ten figures are presented. Five Gain charts and five Return set precision charts. For both types of figures, the first four display the results of the Reuters categories and the fifth displays the results of the client data.

Figure D.1 shows the Gain Chart for the *Strategy / Plans* category. The line for the SVM drawn right underneath the line of the BSVM, which shows almost identical results. The ST-SVM performs best in the first half of the experiment while the TSVM outperforms all other components in the second half of the experiment. The Return set precision chart for the *Strategy / Plans* category can be seen in Figure C.6. Figure C.2 depicts the Gain Chart for the *Regulation / Policy* where the line for the baseline SVM is also drawn underneath the BSVM, once again showing identical results. Figure C.7 displays the return set precision chart. For both the *Strategy / Plans* and the *Regulation / Policy* experiment, the GS-SVM makes up for the extra labelling needed to create the validation set.

The Gain Chart for *Sports*, shown in Figure C.3, displays all components almost identical as a straight line from 0 to 0,9. Only the GS-SVM starts from 8.000 manual labour since the validation set has to be labelled. Figure C.8 depicts the return set precision, which is near 1 almost the entire experiment. The Gain Chart for the *War / Civil War* category is displayed in Figure C.4. The Self-training SVM shows very promising results until iteration 28 (28.000 manually labelled documents), after iteration 28 the TSVM outperforms the Self-training SVM. In the return set precision chart (Figure C.9) it is visible that the return set precision of the TSVM is higher than the ST-SVM.

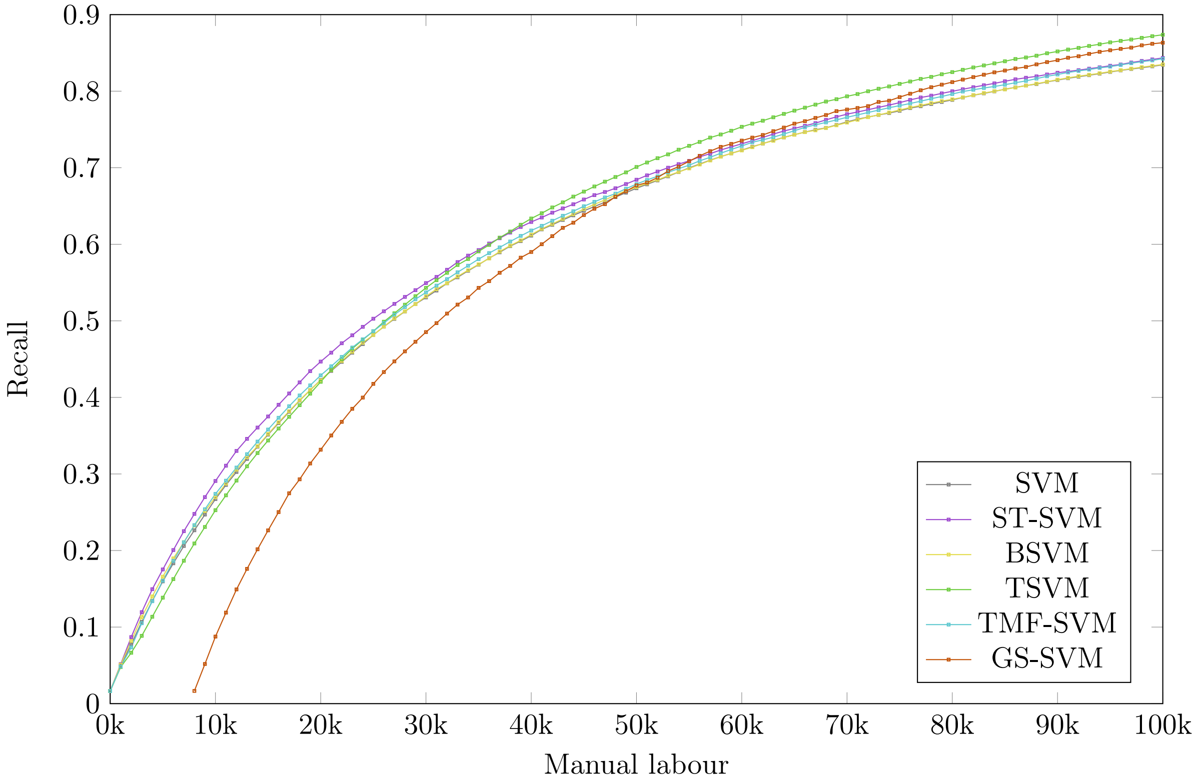


Figure C.1: Gain Chart: Strategy / Plans

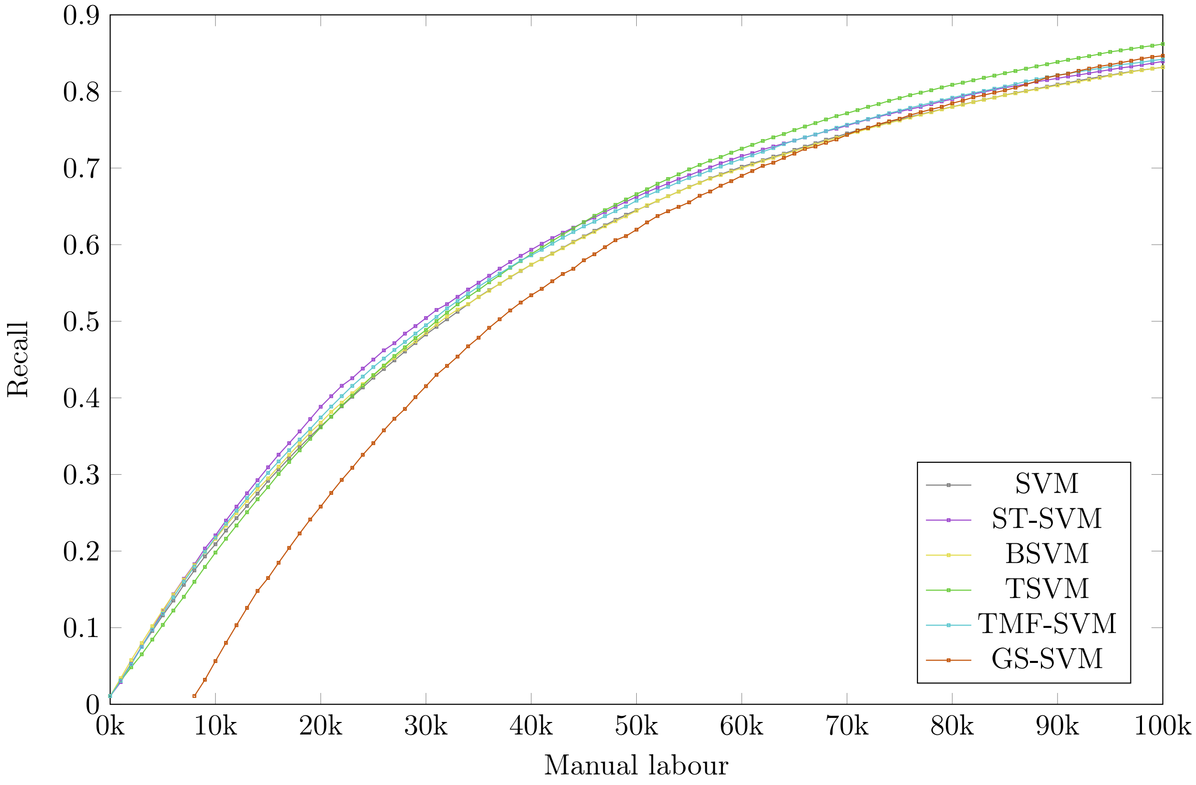


Figure C.2: Gain Chart: Regulation / Policy

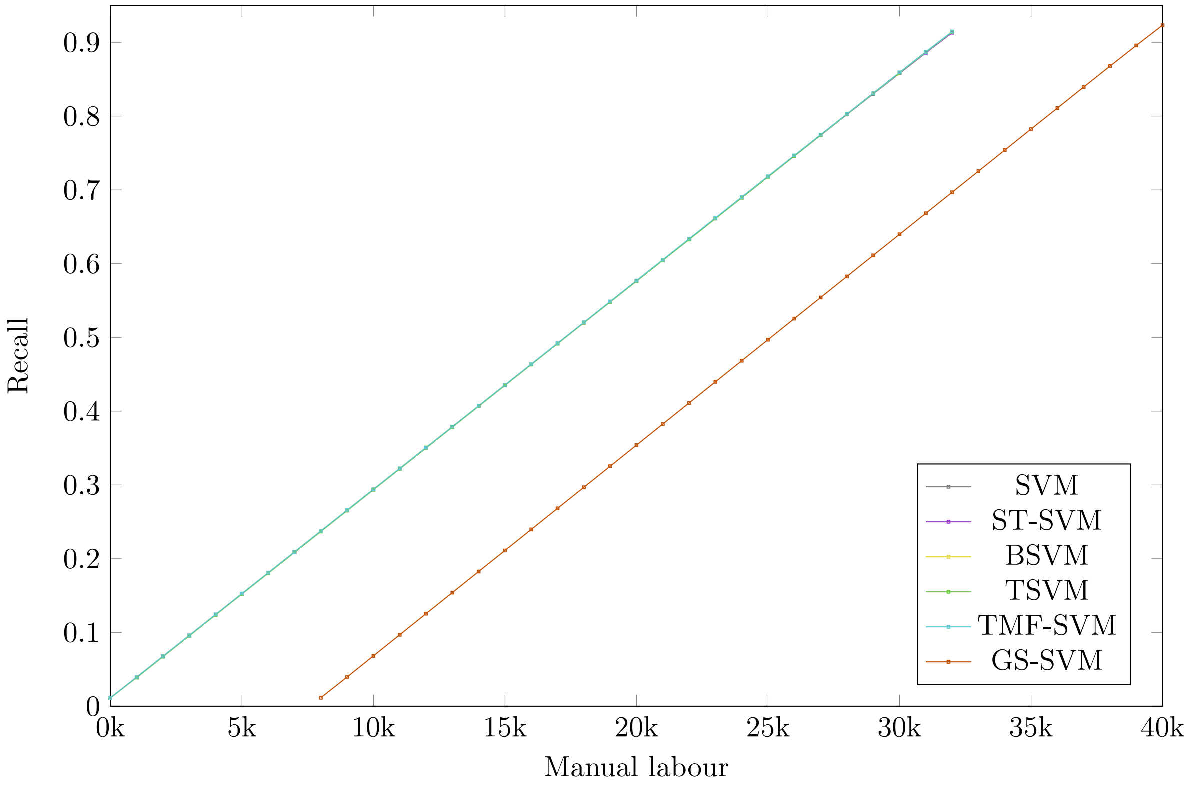


Figure C.3: Gain Chart: Sports

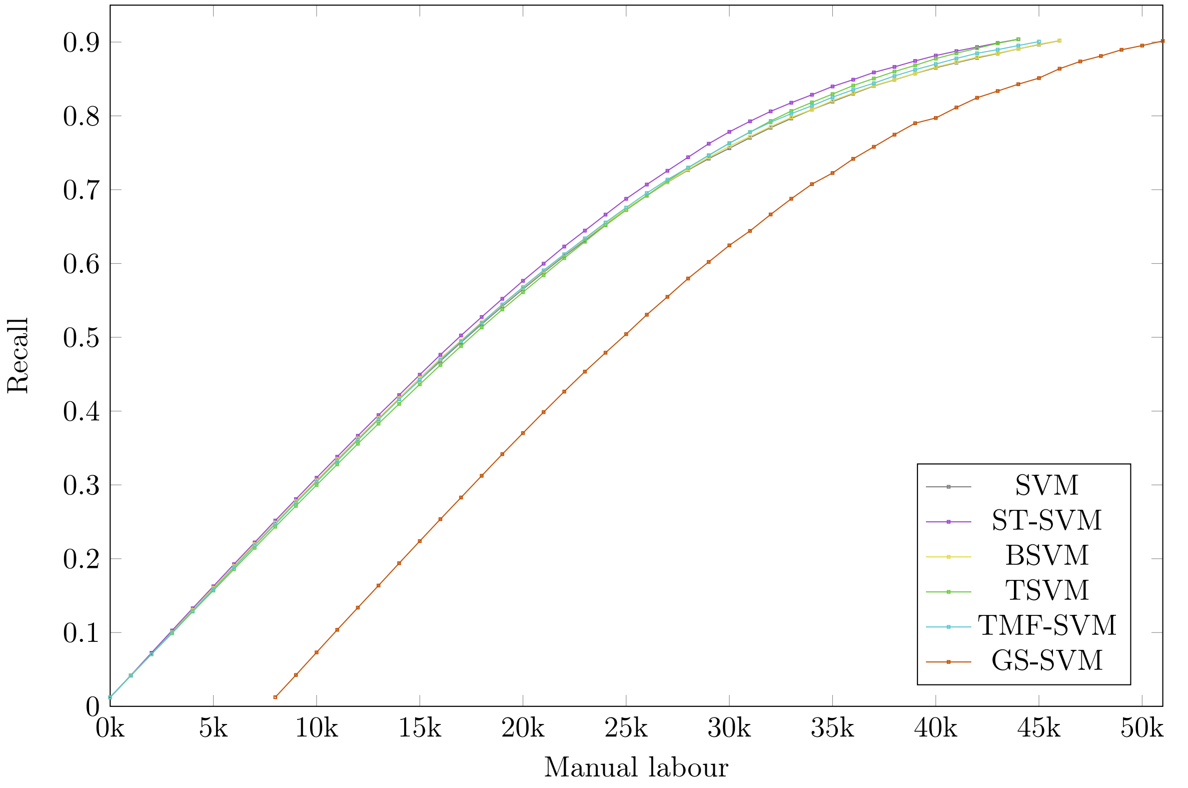


Figure C.4: Gain Chart: War / Civil War

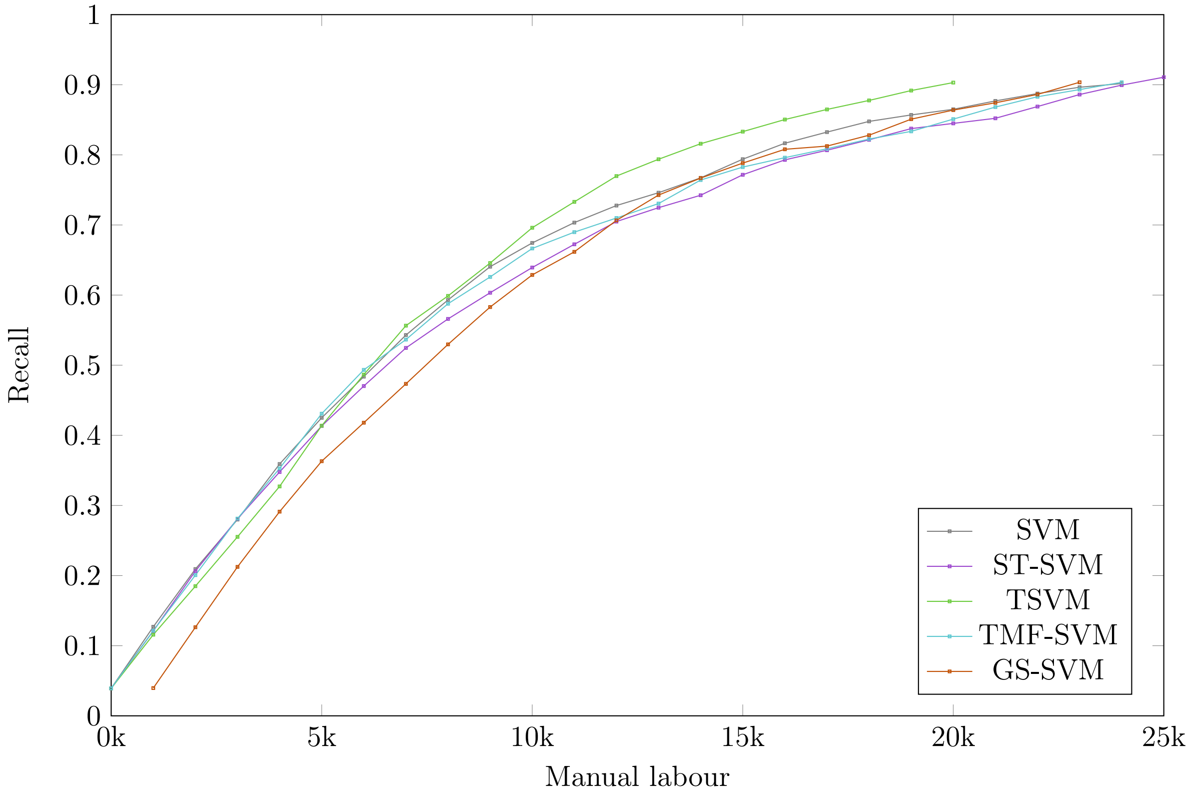


Figure C.5: Gain Chart: Client data

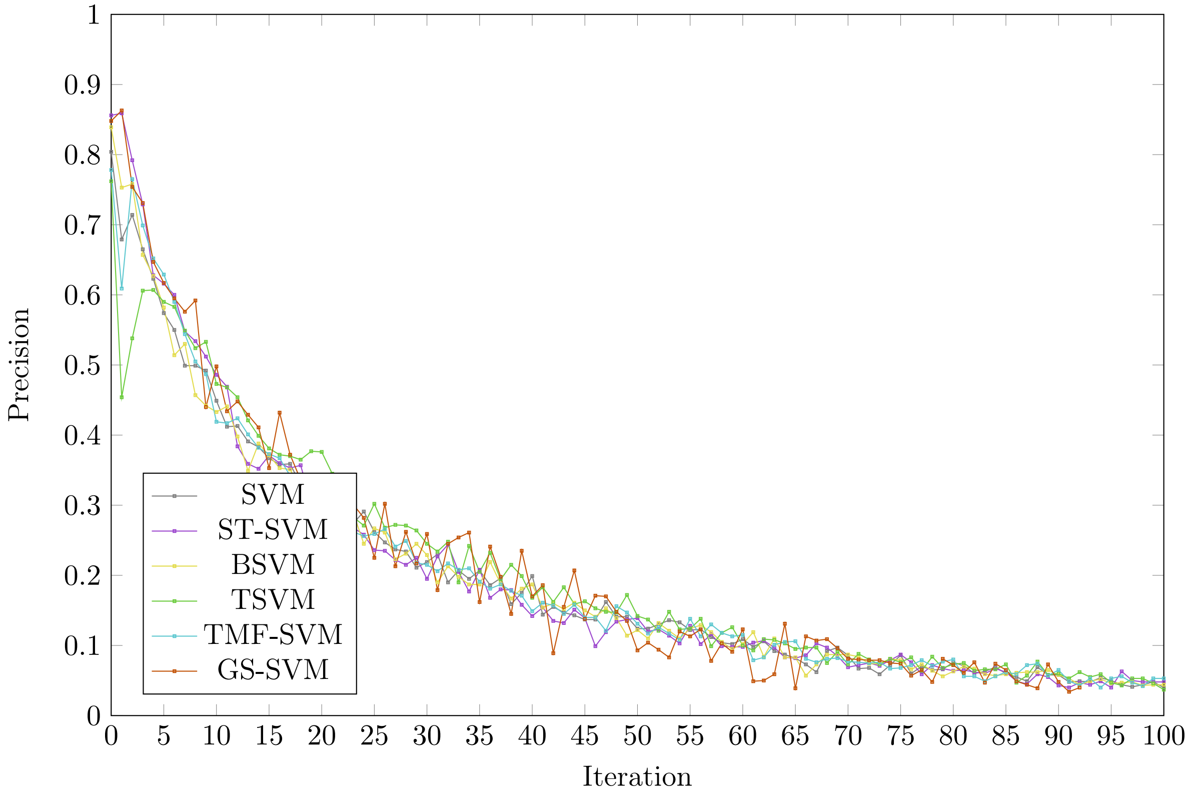


Figure C.6: Return Set Precision Chart: Strategy / Plans

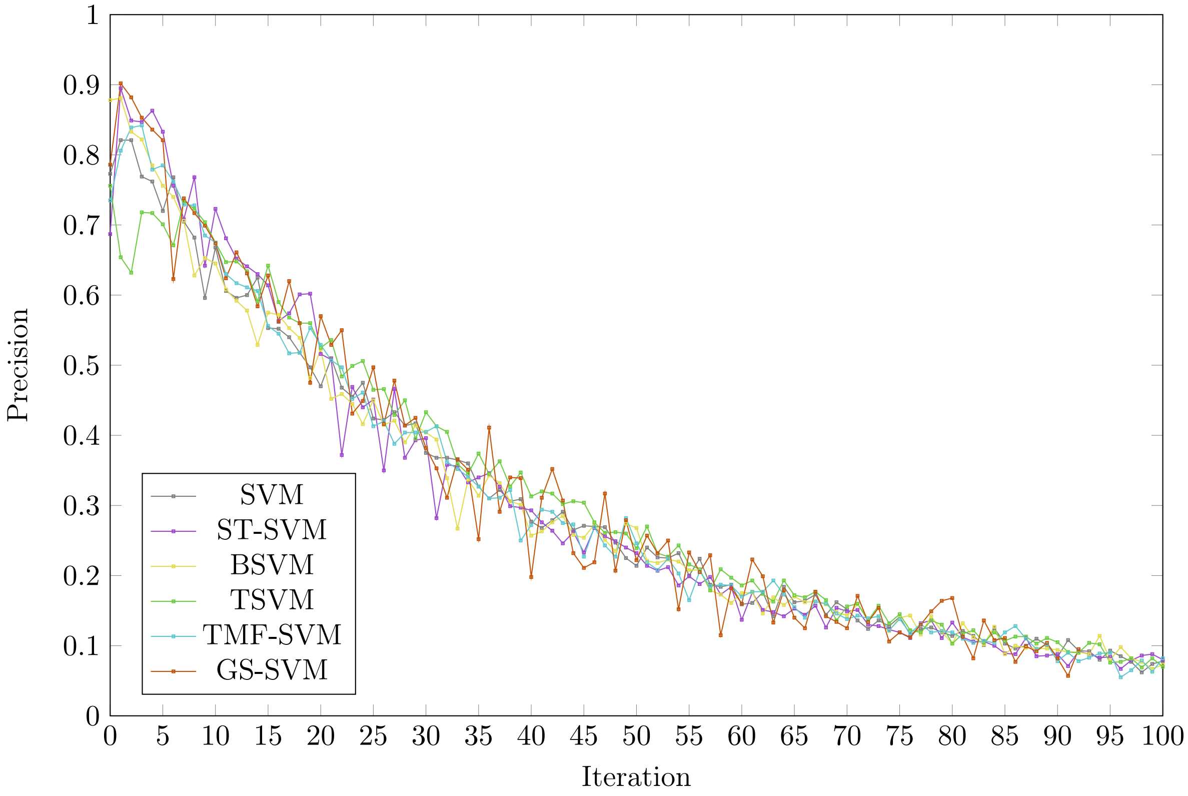


Figure C.7: Return Set Precision Chart: Regulation / Policy

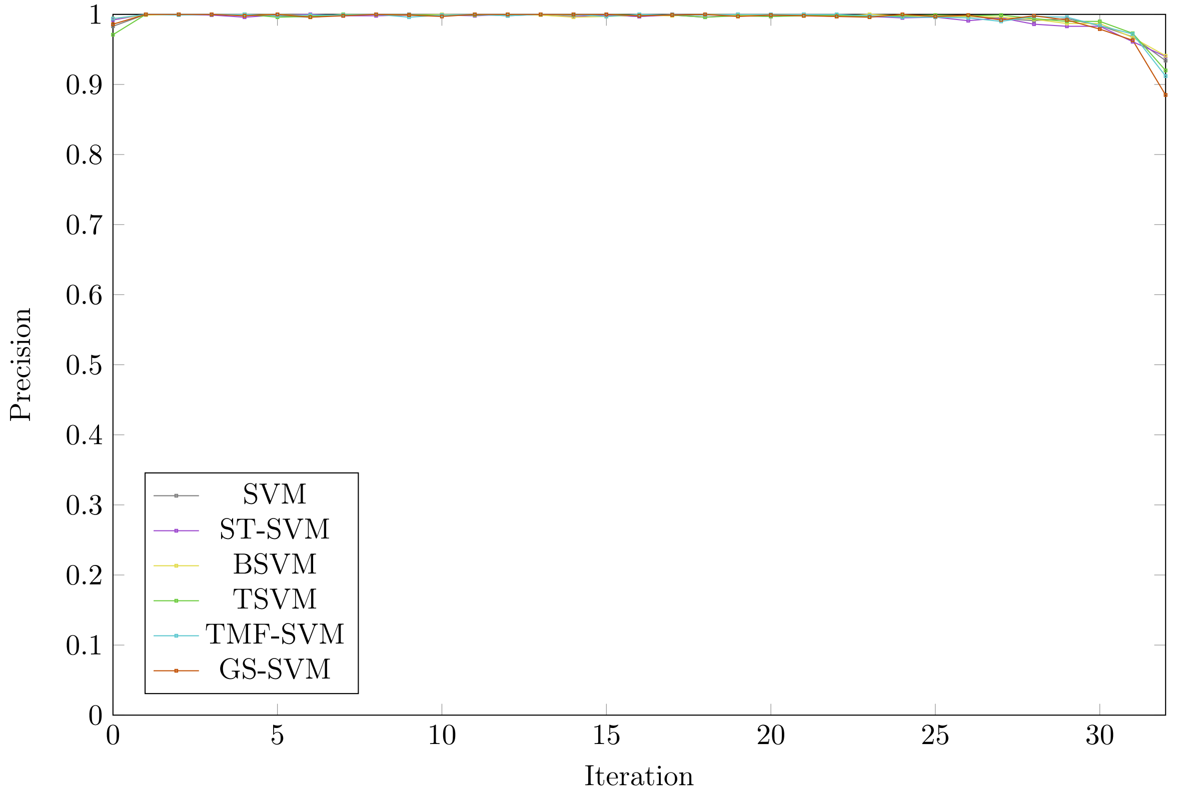


Figure C.8: Return Set Precision Chart: Sports

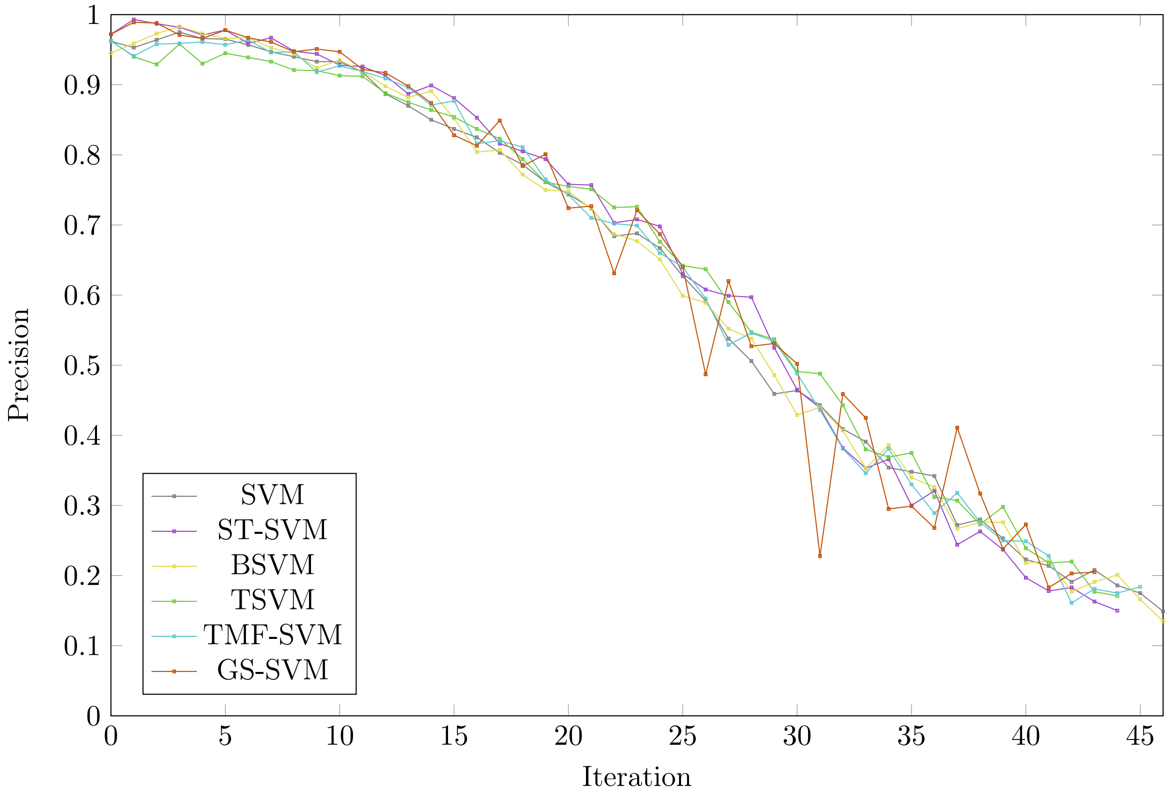


Figure C.9: Return Set Precision Chart: War / Civil War

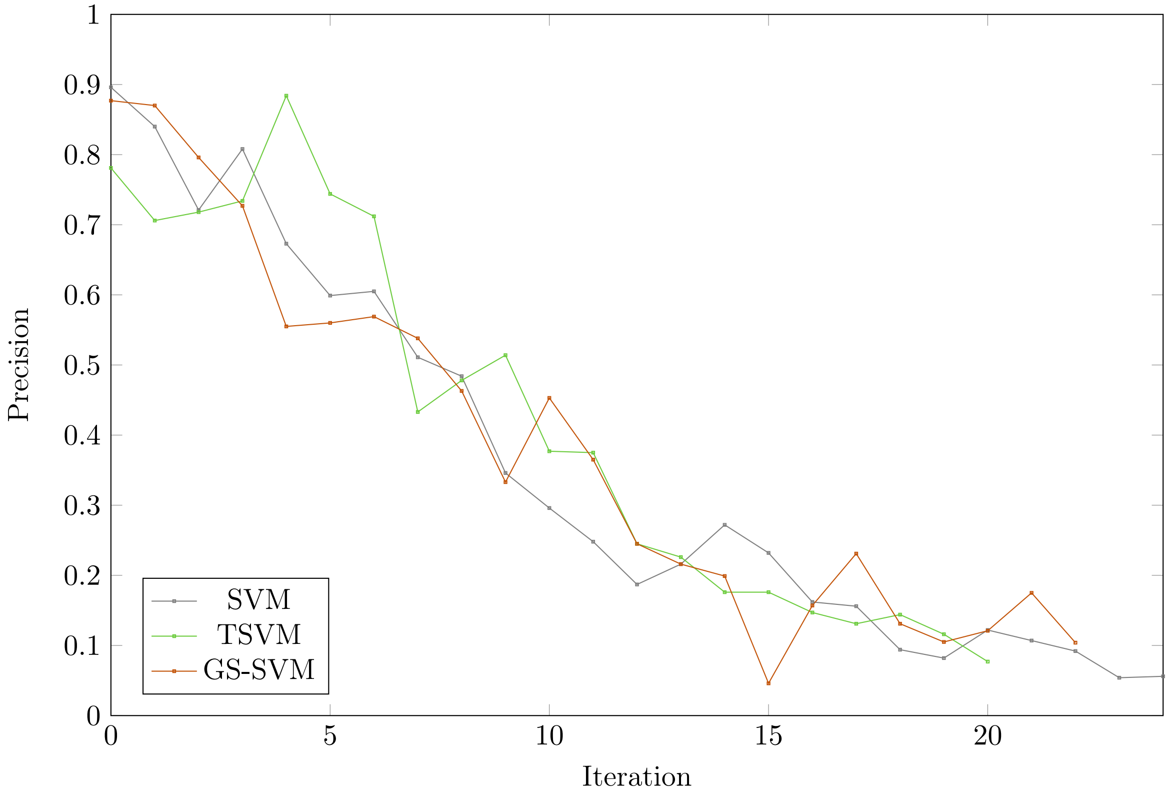


Figure C.10: Return Set Precision Chart: Client data

Appendix D

Sampled Labelling Results

The three tables in this Appendix: Tables D.1, D.2 and D.3, present the Sampled Labelling experiment results for three distinct sets of parameters. For each of the Reuters categories, the number of skipped documents (skipped), the number of False Positives (FP), the gained recall (GR) and the precision of the skipped documents (SP) are shown.

The results shown in Table D.1 are for the experiment with parameters: $t_s = 0.90$ and $t_d = 0.95$. In 63 out of 103 categories, Sampled Labelling caused documents to be skipped during labelling. An average of 21,956 documents are skipped with an average precision of 94.2%.

The results for the experiment with parameters: $t_s = 0.95$ and $t_d = 0.97$ are presented in Table D.2. An average of $24{,}467$ documents are skipped in the 49 categories where Sampled Labelling had an effect. The documents are skipped with an average precision of 0.946.

Table D.3 displays the results for the experiment with the parameters: $t_s = 0.98$ and $t_d = 0.99$. Sampled Labelling has an effect in 43 of the Reuters categories. An average of $21{,}667$ documents are skipped with an average precision of 97.9%. The minimum precision of 94.4% is the highest minimum precision of all three experiments.

Cat	**Skipped**	**FP**	**GR**	**SP**
C11	—	—	—	—
C12	800	71	0.067	0.911
C13	—	—	—	—
C14	—	—	—	—
C15	114700	2734	0.756	0.976
C151	61400	1412	0.750	0.977
C1511	12100	587	0.521	0.951
C152	34200	1373	0.468	0.960
C16	—	—	—	—
C17	13000	835	0.308	0.936
C171	4300	282	0.235	0.934
C172	3100	270	0.270	0.913
C173	—	—	—	—
C174	3900	196	0.664	0.950
C18	15600	1042	0.295	0.933
C181	8700	561	0.201	0.936
C182	—	—	—	—
C183	400	34	0.054	0.915
C21	1000	86	0.039	0.914
C22	—	—	—	—
C23	—	—	—	—
C24	900	98	0.028	0.891
C31	700	60	0.017	0.914
C311	—	—	—	—
C312	—	—	—	—
C313	—	—	—	—
C32	—	—	—	—
C33	1000	65	0.065	0.935
C331	—	—	—	—
C34	100	10	0.021	0.900
C41	5300	228	0.467	0.957
C411	4700	138	0.458	0.971
C42	2000	138	0.168	0.931
CCAT	305400	10502	0.801	0.966
E11	300	25	0.035	0.917
E12	100	9	0.004	0.910
E121	100	1	0.046	0.990
E13	1800	65	0.273	0.964
E131	900	27	0.159	0.970
E132	—	—	—	—
E14	—	—	—	—
E141	—	—	—	—
E142	—	—	—	—
E143	—	—	—	—
E21	20300	967	0.471	0.952
E211	3100	190	0.197	0.939
E212	14800	894	0.540	0.940
E31	200	14	0.083	0.930
E311	200	17	0.118	0.915
E312	—	—	—	—
E313	—	—	—	—
E41	6000	462	0.352	0.923
E411	—	—	—	—
E51	2700	196	0.127	0.927
E511	—	—	—	—
E512	1100	57	0.087	0.948
E513	200	3	0.087	0.985
E61	—	—	—	—
E71	2800	23	0.532	0.992
ECAT	54000	2688	0.450	0.950
G15	9900	663	0.479	0.933
G151	—	—	—	—
G152	—	—	—	—
G153	—	—	—	—
G154	3000	133	0.357	0.956
G155	—	—	—	—
G156	—	—	—	—
G157	—	—	—	—
G158	—	—	—	—
G159	—	—	—	—
GCAT	190900	5420	0.798	0.972
GCRIM	9600	613	0.298	0.936
GDEF	200	16	0.023	0.920
GDIP	8200	601	0.217	0.927
GDIS	2700	146	0.312	0.946
GENT	—	—	—	—
GENV	100	10	0.016	0.900
GFAS	—	—	—	—
GHEA	200	26	0.033	0.870
GJOB	6600	382	0.383	0.942
GMIL	—	—	—	—
GOBIT	—	—	—	—
GODD	—	—	—	—
GPOL	14200	895	0.250	0.937
GPRO	—	—	—	—
GREL	200	7	0.070	0.965
GSCI	—	—	—	—
GSPO	32600	469	0.923	0.986
GTOUR	—	—	—	—
GVIO	9300	565	0.285	0.939
GVOTE	1700	163	0.147	0.904
GWEA	300	9	0.077	0.970
GWELF	—	—	—	—
M11	33600	1791	0.690	0.947
M12	9300	601	0.357	0.935
M13	34800	1810	0.649	0.948
M131	15900	859	0.564	0.946
M132	12000	684	0.449	0.943
M14	68800	2542	0.805	0.963
M141	38800	1702	0.813	0.956
M142	6000	274	0.495	0.954
M143	15300	746	0.697	0.951
MCAT	157100	4180	0.767	0.973

Table D.1: Sampled Labelling results for $t_s = 0.90$ and $t_d = 0.95$

Cat	# Skipped	FP	GR	SP
C11	—	—	—	—
C12	—	—	—	—
C13	—	—	—	—
C14	—	—	—	—
C15	106100	1232	0.699	0.988
C151	58000	1279	0.708	0.978
C1511	10000	277	0.431	0.972
C152	28200	932	0.386	0.967
C16	—	—	—	—
C17	6500	227	0.154	0.965
C171	1900	92	0.104	0.952
C172	1500	57	0.131	0.962
C173	—	—	—	—
C174	2900	82	0.494	0.972
C18	7600	327	0.144	0.957
C181	4600	242	0.106	0.947
C182	—	—	—	—
C183	—	—	—	—
C21	—	—	—	—
C22	—	—	—	—
C23	—	—	—	—
C24	100	5	0.003	0.950
C31	—	—	—	—
C311	—	—	—	—
C312	—	—	—	—
C313	—	—	—	—
C32	—	—	—	—
C33	—	—	—	—
C331	—	—	—	—
C34	—	—	—	—
C41	4100	80	0.361	0.980
C411	4600	136	0.448	0.970
C42	500	25	0.042	0.950
CCAT	283400	6256	0.743	0.978
E11	100	2	0.012	0.980
E12	—	—	—	—
E121	—	—	—	—
E13	1300	35	0.197	0.973
E131	600	11	0.106	0.982
E132	—	—	—	—
E14	—	—	—	—
E141	—	—	—	—
E142	—	—	—	—
E143	—	—	—	—
E21	15700	582	0.364	0.963
E211	400	18	0.025	0.955
E212	11500	372	0.420	0.968
E31	100	1	0.041	0.990
E311	—	—	—	—
E312	—	—	—	—
E313	—	—	—	—
E41	1400	62	0.082	0.956

Cat	Skipped	FP	GR	SP
E411	—	—	—	—
E51	700	34	0.033	0.951
E511	—	—	—	—
E512	500	16	0.040	0.968
E513	100	—	0.044	1.000
E61	—	—	—	—
E71	3500	124	0.664	0.965
ECAT	40600	1666	0.339	0.959
G15	6700	300	0.324	0.955
G151	—	—	—	—
G152	—	—	—	—
G153	—	—	—	—
G154	2000	59	0.238	0.971
G155	—	—	—	—
G156	—	—	—	—
G157	—	—	—	—
G158	—	—	—	—
G159	—	—	—	—
GCAT	182500	3619	0.763	0.980
GCRIM	5100	183	0.158	0.964
GDEF	—	—	—	—
GDIP	2900	110	0.077	0.962
GDIS	1100	31	0.127	0.972
GENT	—	—	—	—
GENV	—	—	—	—
GFAS	—	—	—	—
GHEA	—	—	—	—
GJOB	4400	175	0.255	0.960
GMIL	—	—	—	—
GOBIT	—	—	—	—
GODD	—	—	—	—
GPOL	10200	380	0.179	0.963
GPRO	—	—	—	—
GREL	—	—	—	—
GSCI	—	—	—	—
GSPO	31200	249	0.883	0.992
GTOUR	—	—	—	—
GVIO	6100	298	0.187	0.951
GVOTE	400	30	0.035	0.925
GWEA	—	—	—	—
GWELF	—	—	—	—
M11	29400	1181	0.604	0.960
M12	6300	319	0.242	0.949
M13	31700	1305	0.591	0.959
M131	12500	436	0.443	0.965
M132	9400	371	0.351	0.961
M14	63300	1867	0.741	0.971
M141	35000	1144	0.734	0.967
M142	5000	150	0.412	0.970
M143	13900	503	0.633	0.964
MCAT	143300	2985	0.700	0.979

Table D.2: Sampled Labelling results for $t_s = 0.95$ and $t_d = 0.97$

Cat	# Skipped	FP	GR	SP
C11	—	—	—	—
C12	—	—	—	—
C13	—	—	—	—
C14	—	—	—	—
C15	102000	689	0.672	0.993
C151	53800	584	0.657	0.989
C1511	5500	99	0.237	0.982
C152	20300	485	0.278	0.976
C16	—	—	—	—
C17	2500	64	0.059	0.974
C171	400	12	0.022	0.970
C172	—	—	—	—
C173	—	—	—	—
C174	1900	40	0.324	0.979
C18	1400	46	0.027	0.967
C181	100	2	0.002	0.980
C182	—	—	—	—
C183	—	—	—	—
C21	—	—	—	—
C22	—	—	—	—
C23	—	—	—	—
C24	—	—	—	—
C31	—	—	—	—
C311	—	—	—	—
C312	—	—	—	—
C313	—	—	—	—
C32	—	—	—	—
C33	—	—	—	—
C331	—	—	—	—
C34	—	—	—	—
C41	2700	45	0.238	0.983
C411	2700	25	0.263	0.991
C42	—	—	—	—
CCAT	237700	2810	0.623	0.988
E11	—	—	—	—
E12	—	—	—	—
E121	100	3	0.046	0.970
E13	600	3	0.091	0.995
E131	600	14	0.106	0.977
E132	—	—	—	—
E14	—	—	—	—
E141	—	—	—	—
E142	—	—	—	—
E143	—	—	—	—
E21	10600	227	0.246	0.979
E211	—	—	—	—
E212	10500	205	0.383	0.980
E31	—	—	—	—
E311	—	—	—	—
E312	—	—	—	—
E313	—	—	—	—
E41	1500	84	0.088	0.944
Cat	**Skipped**	**FP**	**GR**	**SP**
E411	—	—	—	—
E51	300	11	0.014	0.963
E511	—	—	—	—
E512	—	—	—	—
E513	100	—	0.044	1.000
E61	—	—	—	—
E71	1400	13	0.266	0.991
ECAT	22000	553	0.183	0.975
G15	2500	75	0.121	0.970
G151	—	—	—	—
G152	—	—	—	—
G153	—	—	—	—
G154	1400	29	0.167	0.979
G155	—	—	—	—
G156	—	—	—	—
G157	—	—	—	—
G158	—	—	—	—
G159	—	—	—	—
GCAT	149200	2153	0.624	0.986
GCRIM	1100	22	0.034	0.980
GDEF	—	—	—	—
GDIP	1000	24	0.026	0.976
GDIS	300	12	0.035	0.960
GENT	—	—	—	—
GENV	—	—	—	—
GFAS	—	—	—	—
GHEA	—	—	—	—
GJOB	1300	10	0.075	0.992
GMIL	—	—	—	—
GOBIT	—	—	—	—
GODD	—	—	—	—
GPOL	3500	51	0.062	0.985
GPRO	—	—	—	—
GREL	—	—	—	—
GSCI	—	—	—	—
GSPO	29900	237	0.847	0.992
GTOUR	—	—	—	—
GVIO	1300	30	0.040	0.977
GVOTE	200	11	0.017	0.945
GWEA	—	—	—	—
GWELF	—	—	—	—
M11	20200	436	0.415	0.978
M12	3100	109	0.119	0.965
M13	21800	454	0.406	0.979
M131	8200	145	0.291	0.982
M132	3500	67	0.131	0.981
M14	54900	802	0.643	0.985
M141	23300	402	0.488	0.983
M142	3200	68	0.264	0.979
M143	10400	194	0.474	0.981
MCAT	112700	1019	0.550	0.991

Table D.3: Sampled Labelling results for $t_s = 0.98$ and $t_d = 0.99$

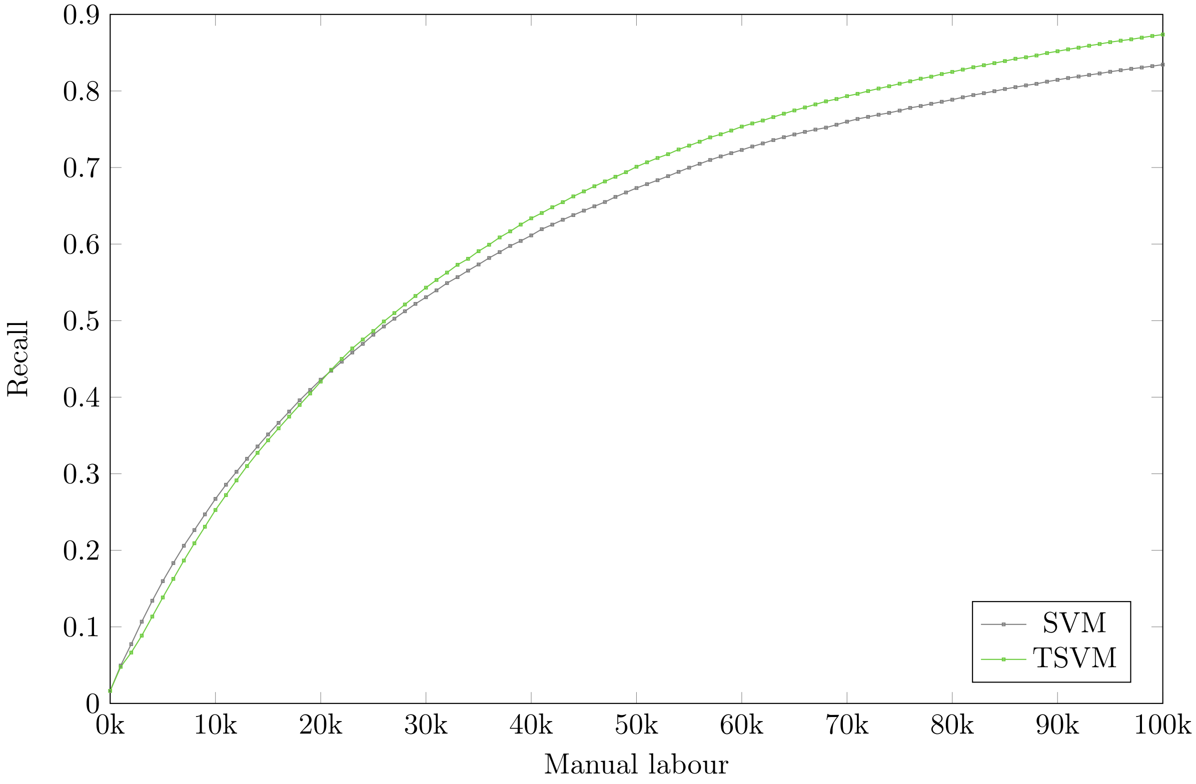


Figure D.1: Gain Chart: Strategy / Plans