

# **Automated Text Categorisation to Create Datasets for Event Studies: An Empirical Study**

Lineke Sneller  
Nyenrode Business University  
l.sneller@nyenrode.nl

2008-08-25  
Submission for IRS AIS 2008

## **Abstract**

This paper studies the feasibility of using automated text categorisation to create the event dataset required for an event study. The machine learning approach for text categorisation was used. Three software packages were trained with a manually created event dataset of 445 ERP-related press releases for the Amsterdam Stock Exchange. After the training, the packages were used to automatically select ERP-related events from 342,922 press releases for the London Stock Exchange.

In total 435 text categorisation experiments have been carried out with varying categorisation indexing parameters. The quality of the experiments was measured using precision and recall. The best experiment resulted in a precision of 4.9% at a recall level of 90.3%. The automated text categorisation reduced the researcher's time required for creation of the event dataset from an estimated 2,058 hours to around 300 hours.

This study shows that automated text categorisation can be used to create input datasets for event studies within reasonable time frames. Automated text categorisation potentially can also speed up the analysis of other textual sources, like open-ended survey questions or interview transcripts.

**Keywords:** automated text categorisation, event study, machine learning

## Introduction

One of the objectives of my research project is to study the effect of ERP implementations on company value for European companies. In order to do this, I want to carry out an event study for two European countries, the Netherlands and the UK. This paper reports on a sub-project of my research project. Subject of this paper is the data gathering method for the event studies.

Event studies are a frequently-used and well accepted research method<sup>1</sup>. They are based on capital market theory. In this theory, share price is considered the measure for company value. In their classic textbook on corporate finance, Brealy & Myers [2000] explain how capital market research establishes this relationship between company value and its share price. A company's share is an asset with two types of future cash flows: dividends and capital gains. Brealy & Myers argue that the share price is exactly equal to the present value of its future cash flows, and therefore, the value of the share is accurately reflected by its price. If the share price reflects company value, then an event that has a significant impact on the share price has a significant impact on the value of the company. This is the basis for the event study research method.

An event study requires three datasets: stock prices for all companies in the study, index prices for the stock markets in the study, and event data. The creation of the first two datasets is relatively straight forward: in countries with mature stock markets stock prices and index prices are readily available at low costs.

The creation of the third dataset can be more complicated. Event datasets can be based on a variety of sources. Often, press releases are the source of events. Most event studies that use press releases use queries on databases like Reuters' Factiva or Lexis Nexis.

Ideally, an event dataset for an event study is complete, it should contain all occurrences of the event. If events that have occurred in reality are not included in the event dataset, their effect on the company's stock price is not taken into account in the event study, which might affect the outcomes of the study.

Completes of course will not be achieved in reality. The next best then is to create event datasets that can be analysed with statistical techniques. In a US context, with large stock markets that have existed for many decades, researchers have been able to create event datasets with relatively simple event selection methods. A few examples of event studies in IT value research based on US event datasets:

- Chatterjee et al. [2001] created an event dataset of 96 newly created CIO positions by key word search of a combination of "new" or "create" with various position titles
- Subramani & Walden [2001] study 251 e-commerce-related events found by key word search of "launch" or "announce" in the same sentence as "online", "e-commerce" or ".com"
- Hayes et al. [2001] found 91 enterprise resource planning (ERP) initiatives by using a combination of the search terms "implement", "convert" or "contract" and the names of the best-known vendors of ERP systems

---

<sup>1</sup> For an earlier research project I searched for event studies published in 2000. I searched for the key words "event study" in the EBSCO online journal article database. I restricted the search to 2000 and to academic peer-reviewed journals, and I found 258 different articles. Of these 258 articles, 54 were based on European datasets [reference to author].

In Europe however, the stock markets for individual countries are often small<sup>2</sup>, and press releases may be issued in several languages. The creation of event datasets by relatively simple queries on press release datasets may lead to event datasets that are too small for event studies<sup>3</sup>.

The objective of the paper is to research whether the creation of an event dataset for several European countries can be carried out using a combination of manual and automated text categorisation. The remainder of this paper has five sections. In the first section, automated text categorisation techniques are briefly described, and the research question of the paper is presented. In the second section, the research hypotheses are developed. The third section contains the experiment design. The fourth section is the presentations of the results. The paper ends with discussion of the results and areas for further research.

I have used various software packages for the text categorisation experiments that I report on in this paper. At this moment, I have not yet shared most of my results with the various suppliers of the software packages. For this reason I have replaced the supplier names by Package1, Package2 and Package3.

## 1. Background and Research Questions

In this section I give a brief overview of the theoretical background of text categorisation. First, I explain some frequently-used terms in this field of expertise, and the process of text categorisation. Then, I describe the main techniques used for text categorisation, as well as quality measures of text categorisation experiments. The section ends with the research question of this paper.

### 1.a Terminology

In the text categorisation domain, different authors often use different terms for almost the same notion. In order to avoid confusion and to present the terminology I will use throughout this paper, I will use this section to describe the most commonly-used term for each notion. Where possible, I will follow the terms used by Sebastiani, a leading author in this domain. I will also define frequently-used synonyms; these are needed for a good understanding of references to literature in the remainder of the text in this paper.

Sebastiani [2002] defines *text categorisation* as the activity of labelling natural language texts with thematic categories from a predefined set. The thematic categories are also called *labels* [Sebastiani 2002], or *classes* [Giorgetti & Sebastiani 2003].

Text categorisation is also known as *document classification* [Sebastiani 1999], *content analysis* [Ahuvia 2001], *text classification* or *topic spotting* [Sebastiani 2002]. The natural language texts are also known as *documents* [Sebastiani 1999] or *focal texts* [Ahuvia 2001].

---

<sup>2</sup> A few examples: the main markets in France, Belgium and The Netherlands have 40, 20 and 25 companies listed concurrently.

<sup>3</sup> I translated the queries used by Chatterjee et al. [2001], Subramani & Walden [2001] and Hayes et al. [2001] to Dutch and ran them on Factiva, which resulted in 0, 0 and 0 events for Dutch listed companies

Text categorisation has several appearances, depending on the situation in which it is used. In *single-label* categorisation, each label has to be assigned to a document a fixed number of times. In *multi-label* categorisation, any number of labels can be assigned to each document. In *binary* categorisation exactly two labels exist, and each document is either assigned to the first label or to its complement [Sebastiani 2002].

The definition of text categorisation implies that this activity requires a predefined set of thematic categories. The process of identifying the prospective categories is called *clustering* [Mostafa et al. 1998]. Categorisation can be carried out in two styles: *document-pivoted* or *category-pivoted*. In the document-pivoted style, for a given document all categories are selected in which the document should be categorised; this style is appropriate when documents are added during the categorisation. In the category-pivoted style, for a given category all documents are selected that should be categorised in it; this style is appropriate when categories are added during the categorisation [Sebastiani 2002].

### **1.b Steps in the text categorisation process**

Text categorisation is a process that can be split into several steps. According to Ahuvia [2001], traditional content analysis is divided into three steps: selection of the focal texts, coding the focal texts and interpreting the results of the coding. In Figure 1, I present a graphical representation of Ahuvia's description of the text categorisation process.

<<< Figure 1 around here >>>

Text categorisation is a subjective task [Giorgetti & Sebastiani 2003]. Ahuvia [2001] describes this characteristic of traditional content analysis in the following way:

"Content analysis is a method for interpreting the meaning of texts [...]. Since meaning exists in people, and people may understand the same text in different ways, researchers face an important issue: whose understanding of the text should be used as the basis for coding? The text's authors? The text's natural readers? The researchers? Or some combination of these?" [Ahuvia 2001]

Nastase et al. [2007] propose that ideally, more than one coder should be involved in the coding. The results of each of the coders can then be compared. Kassanjian [1977] observes that in consumer behaviour literature, this requirement has seldom been met:

"Typically, the author has analysed the communications material himself with no expressed concern about the reliability of the analysis or controls for selective perceptions and biased predispositions." [Kassanjian 1977]

When multiple coders are used for the text categorisation, the process becomes very labour-intensive. According to Nastase et al. [2007], this labour-intensity has restricted qualitative research, as only small samples can be used. The importance of text categorisation however has increased in recent years. The exponential growth of the number of online documents and the increased pace with which information needs to be distributed has created the need for automatic text classification [Joachims 2001].

## 1.c Approaches towards automated text categorisation

The idea of automating text categorisation is not new. Traditionally, two approaches towards automatic text categorisation have existed. Until the late 1980s, the most popular approach to text categorisation was *knowledge engineering*. This approach consists of manually defining a set of rules that encodes expert knowledge on how to classify documents under the given categories [Sebastiani 2002]. The encoded rules are programmed in an *expert system*, a computer program intended to mimic the decisions of a human expert. An expert system provides a recommendation to the user and has the ability to show how and why it reached that conclusion [Leech & Sangster 2002].

Kattan et al. [1993] find that the knowledge engineering approach has not always been successful. They see that in many cases, the rules used by the human expert are not readily discernable by observation or interview. The coding of expert knowledge fails because experts have difficulties in explaining their decision processes, they are typically more confident demonstrating their decision-making process than they are explaining it, and experts may be reluctant to reveal their rules directly. Sebastiani [1999] adds another drawback of expert systems: manual intervention from a knowledge engineer as well as a domain expert is required, in both the initial creation and in the maintenance of the expert system. He also mentions the lack of replicable studies that report the effectiveness of text categorisation with knowledge engineering.

Nowadays, the most used approach to text categorisation is *machine learning*. This approach requires a *corpus*, a limited set of documents that have already been categorised. The corpus is input for the *learner*, a computer program that automatically generates a categorisation rule by observing the corpus, and subsequently uses the rule to automatically categorise *requests*, documents to be classified [Joachims 2001, Sebastiani 1999]. A graphical representation of automated text categorisation with machine learning is presented in Figure 2.

<<< Figure 2 around here >>>

Machine learning resolves the drawbacks of knowledge engineering mentioned above. Firstly, no manual intervention is required after the learning phase. Related to this, considerable savings can be achieved in terms of expert labour power. Lastly, a growing body of empirical research, some of which I will summarise below, reports on the accuracy of learners themselves and their accuracy in comparison to human experts.

Giorgetti & Sebastiani [2003] describe the three phases that are most commonly distinguished in text categorisation with machine learning. Joachims [2001] uses the term *model selection* for these three phases.

The first step is *indexing*: mapping each document in the corpus into a compact representation of its content that can be directly interpreted by the learner. A sub-step of indexing often is *stemming*, replacing conjugations of words by their stems. The main purpose of indexing is increasing the calculation performance of the learner by reducing the dimensions of the documents in the corpus. Stemming, like many other indexing operations, is language-dependent; this implies that automated text categorisation requires all documents to be written in the same language.

The second step is *learning*: tuning the parameters of the learner based on the categorisation of the corpus. For this purpose, the pre-categorised corpus is divided into a *training set* and a *test set*. The training set is used as input to the learner. Based on this input, the learner generates its categorisation rule and applies it to categorise the test set.

The last step in model selection is *evaluation*: measuring the quality of the categorisation of the test set, by comparing it to the pre-categorisation of the same test set. The learning and evaluation phases can be carried out repeatedly to find the best rule, i.e. the set of learner parameters that results in the highest categorisation quality. Many different quality measures exist. Categorisation quality is discussed in more detail in the next section of this paper.

### 1.d Quality measures for text categorisation

Automated text categorisation is promising, because it is less labour-intensive than manual text classification. However, automation is a feasible route to go only if its quality can be measured and is sufficiently high.

The first issue to be discussed here is the quality of text categorisation *in general*, either manual or automated. Kassirjian [1977] argues that text categorisation has to satisfy three criteria when it is used in academic research. Firstly, it has to be objective: each step in the categorisation process must be carried out in such a way that it is replicable. The objectivity requirement gives text categorisation its scientific standard and differentiates it from literary criticism. Secondly, text categorisation has to be systematic: it has to be done according to consistently applied rules. The systematisation requirement is meant to eliminate biased or partial analysis. Lastly, text categorisation has to be quantitative: the results must be amenable to descriptive as well as inferential statistics. The quantification requirement differentiates text categorisation from ordinary critical reading<sup>4</sup>.

Sebastiani [2002] distinguishes two classes of quality measures for automated text categorisation: *effectiveness* and *efficiency*. Effectiveness is a categoriser's ability to take the right categorisation decision. In the machine learning approach, the quality of the generated categorisation rule is measured by comparing its performance on the test set with the manual classification of the test set. Quality measures in automated text categorisation are mostly based on the so-called *contingency table*. In Figure 3, the template for a contingency table is presented.

<<< Figure 3 around here >>>

In each cell of a contingency table, the number of documents in the test set for the respective category is given. Several quality measures have been designed on the basis of the contingency table. In Table 1, I give an overview of those mentioned by [Lewis 1991, Joachims 2001, Sebastiani 2002].

<<< Table 1 around here >>>

---

<sup>4</sup> In my opinion, automated text categorisation has a higher quality than its manual counterpart when scored on Kassirjian's quality criteria. After all, the learner used by automatic categorisation is a computer program that can be run more than once with the same outcome; this guarantees objectivity. Moreover, a computer program consists of coded rules, which implies the systemisation as required by Kassirjian. Lastly, the outcome of automated categorisation is quantitative, so the third criterion mentioned by Kassirjian is also met. As the remainder of this paper is dedicated to automated text categorisation only, I will assume that automated text categorisation satisfies Kassirjian criteria; I will therefore pay no further attention to the quality of *general* text categorisation and restrict the discussion to the second issue: the quality of *automated* text categorisation.

The second class of quality measures comprises efficiency, the time required for a categoriser to classify a document [Sebastiani 2002]. Many categorisers are based on mathematical optimisation techniques. Theoretically, an optimal solution can be found for this type of problems. Practically, computers do not yet have the memory required when large training sets are used [Joachims 1998]. Even if it can mathematically be proven that a certain technique will find the optimal categorisation solution, it is important for all practical applications of text categorisation that the solution is found within finite time. Efficiency can be measured in cpu seconds, number of seconds it takes to carry out the computations on the central processing unit of a computer. Efficiency therefore is computer-specific.

Quality measures for automated text categorisation have to be used with care. The frequently-used measures precision and recall do not make sense in isolation. Often, tuning is carried out, either more liberal (higher recall) or more conservative (higher precision). Therefore, a classifier should be evaluated by a combined measure [Sebastiani 2002]. Moreover, in zero-denominator results, it is unclear what has to be reported [Lewis 1991].

### **1.e Research questions**

In the previous subsections, I gave a brief theoretical background of the text categorisation. I presented definitions, discussed the steps in text categorisation, went through the two approaches of automated text categorisation, and listed quality measures for text categorisation. In this section I present my research question.

In [reference to author], I described an event study I have carried out for the companies list on the Amsterdam Exchange (AEX). An event study requires three datasets: stock prices for all companies in the study, index prices for the stock market in the study, and press releases that report on the event being researched. Stock and index prices are readily available, but press release datasets require careful research and selection.

In order to create the press release dataset required for the AEX study, I carried out a manual text categorisation with two coders. In Table 2, Panel A, the actual time spent on this manual categorisation is presented.

<<< Table 2 around here >>>

The most labour-intensive step of the categorisation, the coding step, was carried out by two coders. The step required 397 hours in total, or an average of 6.00 hours per 1,000 documents. The categorisation led to 445 English documents classified as Yes, or a low proportion of 0.67% of the total dataset.

A replication of this AEX study will be carried out for the FTSE-100 companies, the 100 largest listed companies of the London Stock Exchange (LSE). The first step in this replication has already been completed: the selection of documents using the query already designed for the AEX and applying it to Factiva for the LSE. This led to a document set of 342,922 documents.

The next step would then be to plan the time and manpower required for the categorisation. I used the average of 6.00 hours per 1,000 documents that I found for the AEX, and used linear interpolation to estimate the time and labour requirement for the manual categorisation of the LSE dataset. This led to a time estimate of 2,058 hours or 1.29 full time equivalent (fte). From a cost point of view this manual categorisation would require a considerable investment. From

a time span point of view, manual categorisation with two coders would make me miss the deadline for my research project.

An alternative to manual categorisation could be automated text categorisation. The categorisation of the LSE dataset can be modelled as a binary document-pivoted categorisation problem with classes *Yes* and *No*. An important requirement for the machine learning approach to text categorisation, the availability of a manually categorised corpus, is fulfilled with AEX dataset. However, automated text categorisation is an option only if it can be carried out with high enough quality. I therefore pose the following research questions:

- *R<sub>1</sub>: Can automated text categorisation solve the binary document-pivoted categorisation problem for the LSE dataset with sufficiently high categorisation quality?*
- *R<sub>2</sub>: How can categorisation quality be influenced by model selection?*

In the next section I will use existing empirical literature on text categorisation to develop research hypotheses based on these two research questions.

## **2. Hypotheses**

In this section I will derive quality requirements for my LSE experiment, and then explore whether it is reasonable to expect that automated text classification will satisfy these quality requirements. I first describe quality norms and quality levels found in academic research. I then use these norms and levels to develop research hypotheses based on the research question.

### **2.a Empirical findings for text categorisation quality**

Research on the effectiveness of automated text categorisation is scarce. Lewis & Ringuette [1994] remark that though text categorisation is a task of increasing importance, the bulk of text categorisation research has been conducted by organisations with a pressing operational need. Little is known about the effect that characteristics of the texts have on inductive learning algorithms, or about the performance of purely learning-based methods. Lombard et al. [2004] report a continuing lack of careful reporting of reliability, and encourage researchers to follow generally accepted systematic procedures for assessing reliability. Howland et al. [2006] find few hard and fast rules for acceptable boundary values. They see the absence of an agreed measure for inter-coder reliability as one of the issues in text analysis literature.

The limited empirical research that is available reports on categorisation effectiveness only; I have found no empirical research that describes measured computational efficiency. The available research can be split in three groups. The first group discusses the effectiveness of manual text categorisation, especially those cases in which more than one coder is present. The second group compares the effectiveness of automated text categorisation to that of manual categorisation. The last group evaluates the effectiveness of automated text categorisation, by comparing various learners or model selections. I will summarise each of the three groups separately, with a focus on effectiveness norms and empirically measured effectiveness values.

### *2.a.1 Effectiveness of manual text categorisation*

I found three articles that pay explicit attention to the effectiveness of human coders. Kiecker et al. [2000] investigate gender-related effects on inter-coder reliability. They first collect a dataset of 124 cases written by men and women, and consequently pre-code the cases by two women and one man. They then use a group of 18 men and 17 women to record the cases. The authors use a Chi-square test, and they find significant gender-related differences in the coding.

Carlsson et al. [2001] carry out a study with manual text categorisation in the domain of thoracic surgery. A group of four professional physicians categorise a dataset of 100 patient cases using four different classification schemes that are in use in hospitals. The effectiveness of the coding is again measured via inter-coder reliability. The statistic used however is not a Chi-square, but the generalised kappa, a norm which is often used in medical research. The reliability value varies between 0.72 and 0.78, which is considered good according to the generalised kappa boundary values used by the authors.

Howland et al. [2006] describe an experiment that categorises 90 news paper articles on the ozone hole, to understand the rhetorical structure and tenor of news reporting on this subject. Each article is categorised manually by three coders<sup>5</sup>. Again, inter-coder reliability is used as a measure, but again a different statistic is used. The statistic used here is Cohen's kappa. This measure's calculation differs from the generalised kappa, and it has a scale with five boundary values. An average reliability of 0.55 is found empirically by the authors, which on the scale corresponds to moderate reliability.

I only found three empirical articles that explicitly compare the effectiveness of human coders. In this limited number of articles, three different performance measures are used. I conclude that there is no single, generally accepted measure for assessing the effectiveness of human text categorisation.

### *2.a.2 Comparing effectiveness for manual and automated text categorisation*

Four studies were found in which manual and automated text categorisation are compared. In the first study, Mostafa et al. [1998] compare manual and automated categorisation of documents from the Medline medical database. All Medline documents are categorised manually according to a manually defined clustering scheme called MeSH. For the experiment, a dataset of 7,500 classified documents from 15 existing classes has been selected. The dataset was split into a training dataset of 6,000 documents, and a test dataset of 1,500 documents. The authors use a thesaurus-based categorisation algorithm. They use normalised precision and normalised recall as the measure for categorisation effectiveness. The normalisation involves the assignment of a score for each document in stead of just classifying it as True or False; the normalised efficiency measures take the scores into account. The authors present the precision outcomes only, because they expect normalised recall to follow a similar trend. The normalised precision of the automatic classification on the clustering scheme was between 0.61 and 0.93. These findings do not support a consistent superior effectiveness for either manual or automated categorisation.

Burstein et al. [2002] describe two cases of automation of student essay evaluation. In the first case automated rating of students' essays was studied. In US schools, essays are often rated in a two-step process. In the first step an essay is rated by two raters. If they disagree more than one

---

<sup>5</sup> In this study, the human coders supported by the software package Atlas. This is a package that does not categorise text automatically, but it helps human coders to categorise text manually in an efficient way.

point on a six-point scale, a third rater is asked for a decision. The objective of the study was to automate the first rating step for one rater. An automated learner was trained to rate essays on 270 manually rated essays. The automated rater outperformed manual raters: two human raters agreed in 75% to 80% of the cases, while the combination of an automated and a human rater resulted in a 97% agreement.

Burstein et al.'s [2002] second case describes an automated checker that detects violations of grammar and style rules of English. The checker was trained on a large corpus of texts. Its effectiveness depended on the specific rule being checked. Grammar error detection with a 90% precision for instance lead to an automated recall of 40% to 70% of the errors. F-values for checks of repetition of words showed mixed results when human and automated detection was compared, while human performance measured by F-value was consistently better than automated performance in discourse analysis.

In the last case, which was written by Conway [2006], an analysis is made of how the candidates in a state-wide election in Texas in the US were portrayed in the media. The dataset consisted of 107 press releases in which at least one of the candidates was mentioned. A codebook of 96 clues was designed manually, and then applied to the dataset both manually and automatically. The inter-coder reliability between the human coders was measured with Scott's Pi, and the measured value was 0.86. The outcomes of human and automated categorisation were compared using Spearman's rank correlation test statistic; the outcome of the test was that the categorisations were different. Two main causes of the differences were found. Firstly, human coders missed many of the clues. Secondly, if more than one candidate was mentioned in the text, the automated categorisation often matched clues to the wrong candidate. Conway concludes that both manual and automated categorisation have their strengths and weaknesses.

The empirical findings described above do not give unambiguous support for the superior effectiveness of either automated or manual text categorisation. Additionally, as in the empirical research for manual text categorisation, here again there is no single, generally accepted measure for comparing the effectiveness of manual and automated text categorisation.

### *2.a.3 Effectiveness of automated text categorisation*

Four studies were found in which automated text categorisation is assessed. Lewis & Ringuette [1994] use two existing benchmark datasets that have been used many times by researchers in the text categorisation domain: the Reuters dataset, that consists of 21,450 press releases categorised in 135 financial topic categories, and the MUC dataset, that consists of 1,500 documents categorised in 88 terrorism-related categories. Two machine learning algorithms that are often used in artificial intelligence are compared: a Bayesian algorithm and a Decision Tree. The precision-recall break even point is reported as performance measure. Both algorithms perform better on the Reuters dataset; on this dataset the performance varies from 0.65 to 0.67. Additional tests show that the Decision Tree algorithm performs well at high recall levels: a 0.95 recall still has a 0.28 precision. The authors also remark that model selection in the learning phase has considerable impact on the effectiveness of the categorisation.

Joachims [2001] also uses the Reuters dataset in his research. He adds two more benchmark datasets: WebKB, a collection of 4,183 internet pages classified by function, and the Ohsumed corpus of 50,216 medical abstracts that have been categorised using the MeSH disease categories<sup>6</sup>. The experiments are carried out using a Support Vector Machine (SVM) learner,

---

<sup>6</sup> See also Mostafa et al [1998] discussed in the previous section

of which several parameters are tuned. The precision-recall break even point is used to measure the performance. The author concludes that the results vary per test set, and that there is no single best set of parameters.

Giorgetti & Sebastiani [2003] research automated coding of answers to open-ended survey questions. Their dataset consists of surveys carried out by the US National Opinion Research Center (NORC). The answers to the survey questions have been pre-categorised by the NORC. The authors recategorise 2,197 pre-categorised answers to three of the survey questions, using two dictionary-based expert systems, and two learning algorithms. They use accuracy as the performance measure, and they find accuracies of 0.44 and 0.49 for the expert systems, 0.56 for the Bayesian learner and 0.62 for the SVM learner. They conclude that the machine learning approach outperforms the expert system approach.

The most recent empirical study found was carried out by Nastase et al. [2007]. Their dataset consists of 49 transcripts of electronic negotiation simulations manually split into 5,246 documents. These documents were categorised manually by two experienced coders, and were subsequently used as a corpus for machine learning. Two algorithms have been used in the learning phase: Decision Trees and Naive Bayes. The performance measures used were precision, recall and error. On average, the Decision Tree had the smallest error (0.37).

From these four empirical studies, I conclude the following. Firstly, the measures used to compare the effectiveness of the automated text categorisation are consistent: all four studies report precision, recall or a combination of these measures. However, no guidance is given for boundary values of the measures. Secondly, there is no single best machine learning algorithm that outperforms the others. Lastly, model selection affects the quality of the learner.

#### 2.a.4 Summary

In this section, I summarised the findings with respect to categorisation effectiveness and efficiency of empirical research in the domain of text categorisation.

With respect to categorisation effectiveness, I found that there is no consistent way for measuring this in both manual categorisation research and research that compares manual and automated categorisation. In automated categorisation, the situation is better: in the research I found, effectiveness is measured consistently with combined precision and recall measures. However, even though effectiveness is measured with consistent measures, no norms or boundary values for effectiveness are available for automated categorisation, which means that there is no generally accepted norm for categorisation quality. Moreover, there is not a single best learner, nor is there an optimal set of learner parameters.

With respect to categorisation efficiency, I found no empirical research at all that reports on the efficiency of automated text categorisation.

I therefore conclude that existing empirical research gives only limited guidance in setting up an automated text categorisation experiment.

## 2.b Hypotheses

In this section, I develop two research hypotheses based on the research questions posed above:

- *R<sub>1</sub>: Can automated text categorisation solve the binary document-pivoted categorisation problem for the LSE dataset with sufficiently high categorisation quality?*
- *R<sub>2</sub>: How can categorisation quality be influenced by model selection?*

Firstly, I will develop a hypothesis related to  $R_1$ . Effectiveness of automated text categorisation will be measured by recall and precision, as is customary in the empirical research summarised above. The trade-off between the two measures is made in the following way.

In Table 2 above, it is shown that in the AEX dataset the proportion of documents in the Yes class is very low: 0.67%. This means that the Yes class in the AEX dataset is *sparse*: it has a low number of occurrences in the total document dataset. I assume that the same holds for the Yes class in the LSE dataset<sup>7</sup>.

As the Yes class is sparse, it is important to select as many of the documents in the Yes class as possible during the automated categorisation. In other words, it is important to have a low number of False Negatives for the Yes class. In terms of effectiveness, this means that a liberal approach is required, i.e. a high recall should be achieved. As there are no uniform guidelines for boundary values, I arbitrarily set the target for recall on 90%.

Using Figure 3, it is easy to see that higher recall will go at the expense of precision. A low precision will lead to many False Positives, documents that are classified as Yes, but in fact should be in class No. All False Positive documents will require a manual reclassification to check whether they really represent a Positive. In Table 2 above it is shown that reading 66,176 documents in the AEX dataset required a time span of 36 weeks. With automated text classification, I want to carry out this required manual reclassification within an arbitrary time span of three months, or approximately 13 weeks. This means that the number of Positives in the LSE dataset should not exceed  $13 / 36 * 66,176 = 23,896$  documents, or 6.9% of the total number of 342,992 documents in the dataset. I therefore pose the following hypothesis:

*H<sub>1</sub>: Automated text categorisation can be applied to solve the binary document-pivoted categorisation problem for the LSE dataset with a recall of at least 90% on the test dataset, and a number of positives of no more than 24,000 documents of the request dataset.*

Secondly, I will develop hypotheses related to  $R_2$ . The empirical research of automated text categorisation does not give definitive guidance on how to model a text categorisation in order to achieve the best possible categorisation effectiveness. The various studies do not find a learner that is superior to other learners, nor do they find a single best set of learning parameters. The only consistent finding is that model selection does affect categorisation quality. In line with this finding of earlier research I pose the following hypothesis:

*H<sub>2a</sub>: Model selection has a significant effect on categorisation effectiveness.*

Although the main reason for using automated text categorisation is its expected efficiency when compared to manual categorisation, I have not found any empirical research that systematically<sup>8</sup> reports on the efficiency of automated text categorisation. My last hypotheses therefore will test intuitive assumptions:

*H<sub>2b</sub>: Model selection has a significant effect on categorisation efficiency.*

*H<sub>2c</sub>: Categorisation effectiveness and categorisation efficiency are significantly and negatively correlated*

---

<sup>7</sup> I base this assumption on the fact that the companies Shell, Unilever and Reed Elsevier, that comprise 27.8% of the AEX dataset are also listed on the LSE.

<sup>8</sup> The best I have found was Joachims [2001], who says that his algorithms are more computationally efficient than earlier algorithms.

In the next section, I will describe the design of the text categorisation experiment that I use to test these hypotheses.

### **3. Experiment design**

In this section I describe the design of an automated text categorisation experiment, using the description of steps in such an experiment depicted in Figure 2 above. I first give an overview of the corpus and request datasets, and then describe how the corpus is split in test and training datasets. After that, I give a brief overview of the learners that have been selected. The section ends with an overview of the model selection steps.

#### **3.a Corpus and request datasets**

All documents in the experiment are press releases extracted from Reuter's Factiva; the exact selection process has been described in [reference to author]. The corpus for the text categorisation experiment consists of 66,176 English documents in the dataset for the AEX experiment described in [reference to author]. The request dataset is the dataset of 342,992 English documents for the 168 companies that were listed on the LSE between January 1st 1995 and December 31st 2005.

#### **3.b Training and test datasets**

The corpus has been split in three different ways to create test and training datasets. The first split, named left-right, put half of the dataset into the training dataset, and the other half into the test dataset. The second split, named major-minor, put eighty percent into the training dataset, and twenty percent into the test dataset. The last split, named cod1-cod2, consisted of a training dataset of the documents classified by one of the coders, and of a test dataset with the documents classified by the other coder. In Table 3, the distribution of documents over the datasets is presented.

<<< Table 3 around here >>>

#### **3.c Software packages and learners**

Many software packages are available for automated text categorisation. Three of these packages have been used to test the hypotheses. An overview of the selection process of these three packages is given in Appendix A.

The first package used is Package1, a package for data mining and machine learning that has a plug-in for automated text categorisation [supplier reference removed]. I downloaded version 4.0 of the software from [supplier reference removed]. The learner provided by Package1 for automated text categorisation is called [supplier reference removed].

The second package is Package2, a software program specifically designed for automated text categorisation. Like Package1, the program is available under an open source software licence. I downloaded the software from a website [supplier reference removed]. The learner in the software is called Package2; it has been developed for a research project and is described in [supplier reference removed].

The last package is Package3. The package uses a learner that processes documents in two phases: a multi-lingual natural language processing phase, and a language-independent statistic concept modelling phase that matches patterns [supplier reference removed].

### **3.d Model selection**

#### *3.d.1 Indexing*

In order to test the hypotheses  $H_{2a}$ ,  $H_{2b}$  and  $H_{2c}$ , various indexing parameters have been used in the experiments with the open source packages. The parameters and their values are presented in Table 4. The three stemmers, one word filter, four types of term weighting and six values for minimal term length lead to  $3 \times 4 \times 6 = 72$  parameter combinations for both Package1 and Package2.

<<< Table 4 around here >>>

In addition to the EnglishStopWords parameter in Table 4, I used an additional set of stop words for the Package1 and Package2 experiments. As the names of the ERP vendors and products were used to select the press releases, these names will appear in each document, and can therefore be assumed not to be distinctive for the categorisation task at hand. By putting them in the stop words list they will be ignored by the text categorisation software. The list of additional stop words is presented in Table 5.

<<< Table 5 around here >>>

Package3 recommends not to change parameter settings without advice from their consultants. For this reason, the default versions were set for all parameters.

#### *3.d.2 Learning*

For each of the three learners, the default values for the learning parameters were used.

### **3.e Hardware**

I used a laptop computer with Intel™ Core™2 Duo CPU X86 Family 6 at 1.8 Ghz processor and 1 GByte internal memory to carry out the tests of Package1 and Package3. I used a desktop computer with Intel® Pentium® 4 CPU 2.40 GHz processor and 2.42 GHz, 1.00 GByte of internal memory to carry out the tests of Package2.

### 3.f Measures

For each run, the precision, recall, F-value are presented. The F-value is calculated with  $\beta=4$ , which gives a higher weight to recall than to precision. With these measurements, hypothesis  $H_1$  can be tested directly.

For the Package1 experiments, the efficiency has been measured in CPU seconds; the other two software packages have no options for measuring the efficiency. In order to test the hypotheses  $H_{2a}$ ,  $H_{2b}$  and  $H_{2c}$  regression models have been developed.

### 3.g Regression models

For testing the hypotheses  $H_{2a}$ ,  $H_{2b}$  and  $H_{2c}$ , linear regression is used. In Table 6, the variables in the regression models are explained.

<<< Table 6 around here >>>

The F and SECONDS variables are the measures of the quality of the automated text categorisation run. F represents the effectiveness and SECONDS represents the efficiency. They will be used as dependent variables in the regression models.

The other variables represent the options for corpus selection and indexing described in Tables 3 and 4 above. As only one option has been used for Wordfiltering, this variable has not been included in the model. Three regression models will be estimated:

$$M_{2a}: \quad F = C + \alpha_1 \times \text{CORPUS\_LR} + \alpha_2 \times \text{CORPUS\_MM} + \alpha_3 \times \text{STEMMER\_LV} + \alpha_4 \times \text{STEMMER\_PT} + \alpha_5 \times \text{TERMW\_BO} + \alpha_6 \times \text{TERMW\_TF} + \alpha_7 \times \text{TERMW\_TO} + \alpha_8 \times \text{MINCHAR} + \varepsilon$$

$$M_{2b}: \quad \text{SECONDS} = C + \gamma_1 \times \text{CORPUS\_LR} + \gamma_2 \times \text{CORPUS\_MM} + \gamma_3 \times \text{STEMMER\_LV} + \gamma_4 \times \text{STEMMER\_PT} + \gamma_5 \times \text{TERMW\_BO} + \gamma_6 \times \text{TERMW\_TF} + \gamma_7 \times \text{TERMW\_TO} + \gamma_8 \times \text{MINCHAR} + \varepsilon$$

$$M_{2c}: \quad F = C + \delta_1 \times \text{SECONDS} + \varepsilon$$

The hypotheses  $H_{2a}$ ,  $H_{2b}$  and  $H_{2c}$  will be tested by examining the significance of the coefficients in the vectors  $\alpha$ ,  $\gamma$ , and  $\delta$ .

## 4. Results

In this section the results of the experiments with the three software packages are presented, and the research hypotheses are evaluated.

### 4.a Results for $H_1$

In Table 7, the results of the experiments are presented for the two software packages Package1 and Package2. Each row in the table represents two runs: one for Package1 and one

for Package2. For each run, the precision, recall and F-value are presented. The F-value is calculated with  $\beta=4$ , which gives a higher weight to recall than to precision.

<<< Insert Table 7 around here >>>

Package1 produces many invalid runs: 32 of the 216 (14.8%) runs ended in a zero recall. The average precision (60.7%), recall (8.2%) and  $F_{\beta=4}$  (8.6%) values are much lower than those reported in other empirical research. None of the runs ended in a recall value of 90% or higher. In all valid runs, Package1 ended with a higher precision than recall.

Package2 produces a lower number of invalid runs: 4 of the 216 (1.8%) runs ended in a zero recall. The average precision (0.6%) and  $F_{\beta=4}$  (7.7%) values are lower than those found for Package1, but the average recall (43.8%) is much higher. In 18 of the 216 (8.3%) of the runs, Package2 ended with a recall value of 90% or higher; the highest precision of these 18 runs was 0.8%. In 5 runs (2.3%), Package2 ended with a higher precision than recall.

In Table 8, the results of the experiments are presented for the Package3 software package. Three runs have been carried out, each with a different corpus. A standard set of indexing parameters has been used. For each of the three resulting runs, the precision, recall and F-value are presented.

<<< Insert Table 8 around here >>>

All Package3 runs ended in valid results. The average precision (4.5%) and recall (90.5%) lead to an average  $F_{\beta=4}$  (42.4%), that is much higher than the  $F_{\beta=4}$  values found with Package1 and Package2. In each of the three runs, Package3 ended with a recall value of 90% or higher; the highest precision of these 3 runs was 4.9%.

In testing  $H_1$ , the first step is to find all runs that have a recall of at least 90% on the test dataset. In 18 of the Package2 runs and 3 of the Package3 runs, this recall percentage has been found. In Figure 4, these runs are in the top left corner of the graph.

<<< Insert Figure 4 around here >>>

The second step in testing  $H_1$  is based on the assumption that the run with the highest precision on the test set will also have the highest precision on the request set. The run with the highest precision is the Package3 run for the corpus cod1-cod2. I used this run to categorise the 342,922 documents of the LSE dataset. The software package categorised 20,928 document as Positives for the Yes class, which is below the maximum of 24,000 documents set in  $H_1$ .

The text categorisation experiments presented in this paper show that automated text categorisation can be applied to solve the binary document-pivoted categorisation problem for the LSE dataset with a recall of 90.3% on the test dataset, and a number of positives of 20,928 documents in the request dataset. The experiments therefore support hypothesis  $H_1$ .

## 4.b Results for $H_2$

The hypotheses  $H_{2a}$ ,  $H_{2b}$  and  $H_{2c}$  are tested with the results of the Package1 software only. The other two packages do not have options to time the runs, which means that their efficiency

cannot be measured. Efficiency measurements are required for  $H_{2b}$  and  $H_{2c}$ , and I consider testing for effectiveness for  $H_{2a}$  only less relevant. The results are presented in Table 9. Average effectiveness measured by F-value is 8.6%. The average calculation time per run is 1,359.8 seconds.

<<< Table 9 around here >>>>

The regression models described in section 3.g above have been tested with ordinary least squares regression. The resulting coefficients are presented in Table 10.

<<< Table 10 around here >>>>

The first regression model,  $M_{2a}$ , tests whether model selection has a significant effect on categorisation effectiveness. The coefficients for the parameters in the regression model are all significant. Hypothesis  $H_{2a}$  is therefore supported by the data: categorisation effectiveness is affected significantly by model selection for all four parameters: corpus selection, stemming, term weighting and minimum term length.

A more detailed analysis of each of the four parameters shows the following. Firstly, no consistent pattern can be derived of how corpus selection affects effectiveness. No direct influence of the size of the training and test datasets was found. In the experiments, the smallest training dataset is the left-right split, while the largest test dataset is the major-minor split. Additional analysis (regression results not presented) shows that the cod1-cod2 corpus has the highest influence on effectiveness. Secondly, with respect to stemming the experiments show that the Lovins stemmer has the highest effect on categorisation quality, followed by the Porter stemmer. Thirdly, for term weighting, TFIDF outperforms the other three term weighting methods. Lastly, the higher the minimal term length, the lower the categorisation effectiveness.

The second regression model,  $M_{2b}$ , tests whether model selection has a significant effect on categorisation efficiency. The coefficients for the corpus parameters in the regression model are all highly significant, while all term weighting parameters are not significant. The stemmer parameters show a mixed picture, while the minimal term length has a significant and negative effect on categorisation efficiency. I therefore conclude that hypothesis  $H_{2b}$  is only partly supported by the data: categorisation efficiency is affected by some of the model selection parameters.

The last regression model,  $M_{2c}$ , tests whether categorisation effectiveness and efficiency are related; the hypothesised relationship is negative. The coefficient in this regression model is not significant, implying that there is no significant relationship between categorisation effectiveness and efficiency. Hypothesis  $H_{2c}$  is rejected.

## 5. Discussion

### 5.a Summary of the findings

The first research question of this paper was:

- *R<sub>1</sub>: Can automated text categorisation solve the binary document-pivoted categorisation problem for the LSE dataset with sufficiently high categorisation quality?*

Three software packages have been used to carry out a total of 435 experiments to find the answer to this research question. The experiments show that the question can be answered affirmatively. This affirmation implies that I can extend an event study carried for the AEX with a replication for the LSE within the time frames of my research project.

The second research question of this paper was:

- *R<sub>2</sub>: How can categorisation quality be influenced by model selection?*

Categorisation quality has two aspects: effectiveness and efficiency. One of the three software packages has been used to carry out 184 valid experiments with a variation over four classes of model selection parameters to find the answer to this research question.

It was found in the experiments that model selection affects both effectiveness and efficiency. For two of the four parameter classes, i.e. corpus selection and stemming, no consistent influence could be derived that directly affected categorisation quality.

For the parameter class term weighting, TFIDF showed a significant positive effect on categorisation effectiveness and the effect on categorisation efficiency was not significant. Categorisation quality could therefore be influenced positively by using TFIDF for term weighting.

Increasing the minimum number of characters in a term has a negative effect on categorisation effectiveness and positive effect on categorisation efficiency. A quality trade-off can be made using this parameter in model selection.

## **5.b Relevance for research**

The results of this study can be used in various other research domains. Firstly, I created a sizeable corpus of 66,176 documents that can be used for benchmarking text categorisation algorithms. Currently, researchers often refer to the Reuters, MUC or MesH datasets; my ERP dataset could be reused for benchmarking purposes.

Secondly, this study shows that automated text categorisation can be used to create input datasets for research that is based on text datasets within reasonable time frames. Automated text categorisation can speed up the data gathering phase of event studies, research with open-ended survey questions and similar types of research with acceptable quality.

Lastly, this study shows a meaningful way to test inter-coder reliability in sparse datasets that have been created with manual text categorisation. In sparse datasets, the often-used inter-coder reliability measures show high numbers even if the coders disagree on the positive examples. If the results of one coder are used as a training set, and the results of another coder are used as test set, automated text categorisation can be used to assess the categorisation quality.

## **5.c Relevance for practice**

Automated text categorisation can be relevant for all environments where large amounts of documents need to be processed. A practical situation in which automated text categorisation could be used is automated answering of frequently asked questions to a help desk. Initially, a

corpus could be built by manually classifying each incoming request into classes with frequently asked questions, for which standardised answers are created. Regularly, automated text categorisation could be used to assess the quality of the categorisation. Once this quality has become high enough, automated text categorisation can be used to classify new incoming requests and find the answers for these requests.

#### **5.d Limitations and further research opportunities**

This study has a number of limitations. Firstly, the experiments have been limited to the English language only, because some of the model selection parameters that have been used, like stemming and stop word filters, are language dependent.

An opportunity for further research would be an extension of the AEX and LSE event study to the Belgian stock exchange, the BEL-20. The required corpus of Dutch documents is already available: 12,403 Dutch press releases have been categorised manually in the AEX event study. This would also require a text categorisation learner that can be used on Dutch documents. I have not yet found a Dutch learner.

A second limitation of this study is the fact that the experiments are based on corpora consisting of 66,176 ERP-related documents. The manual text categorisation required for the creation of corpora of this size is prohibitive for quick additional research.

In this study, the relationship between corpus split and categorisation effectiveness has been analysed. No significant relationship between the size of the test and training sets on the one hand, and the categorisation effectiveness on the other hand was found.

An opportunity for further research would therefore be to research whether the same or higher categorisation effectiveness can be achieved with smaller corpora. If this could be done, then carrying out event studies with automated text categorisation would become less time consuming, thus allowing many interesting event studies for IT related trends, like outsourcing, structured object architectures (SOA), e-commerce etc.

The last of the limitations of this study I will mention here is the fact that in my experiments automated text categorisation has been used for binary text categorisation only. In the event studies described in [reference to author], this is only the first of two categorisation steps. In the second categorisation step, the Positive class is split into additional categories.

An opportunity for further research would be an attempt to also automate the further categorisation of the Positive class. Though carrying out this second step manually has not been very time consuming for the study at hand, automating it would deepen the understanding of automated text categorisation and the applicability of the technique for multi-label categorisation problems.

## **References**

- Ahuvia, A. (2001). *Traditional, Interpretative, and Reception Based Content Analyses: Improving the Ability of Content Analysis to Address Issues of Pragmatic and Theoretical Concern*. *Social Indicators Research*, 54 ((2001)), 139-172.
- Brealy, R. & Myers, S. (2000). *Principles of corporate finance*. Maidenhead UK: McGraw-Hill.
- Burstein, J., Chodorow, M. & Leacock, C. (2002). *Criterion Online Essay Evaluation: An Application for Automated Evaluation of Student Essays*. (n.a.) Princeton NJ: Educational Testing Services.

- Carlsson, M., Lufström, L. & Hans .Alfeldst, H. (2001). *Classification of Procedures in the Domain of Thoracic Surgery - A Study of Reliability in Coding*. Journal of Medical Systems, 25 (1), 47-61.
- Chatterjee, D., Richardson, V. & Zmud, R. (2001). *Examining shareholder wealth effects of announcements of newly created CIO positions*. MIS Quarterly, 25 (1), 43-70.
- Conway, M. (2006). *The Subjective Precision of Computers: A Methodological Comparison with Human Coding in Content Analysis*. Journalism & Mass Communication Quarterly, 83 (1), 186-200.
- Giorgetti, D. & Sebastiani, F. (2003). *Automating Survey Coding by Multiclass Text Categorisation Techniques*. Journal of the American Association for Information Science and Technology, 54 (14), 1269-1277.
- Hayes, D., Hunton, J. & Reck, J. (2001). *Market reaction to ERP implementation announcements*. Journal of information systems, 15 (1), 3-18.
- Howland, D., Larsen Becker, M. & Prelli, L. (2006). *Merging Content Analysis and the Policy Sciences: A System to Discern Policy-Specific Trends from News Media Reports*. Policy Sciences, 39 ((2006)), 205-231.
- Joachims, T. (1998). *Making Large-Scale SVM Learning Practical*. In Schölkopf, B., Burges, C. & Smola, J. (Eds.), *Advances in kernel Methods - Support Vector Learning*, (pp. 41-56). Cambridge: MIT Press.
- Joachims, T. (2001). *Learning to classify text using support vector machines*. Boston: Kluwer Academic Publishers.
- Kassarjian, H. (1977). *Content Analysis in Consumer Research*. Journal of Consumer Research, 4 (1977), 8-18.
- Kattan, M., Adams, D. & Parks, M. (1993). *A Comparison of Machine Learning with Human Judgment*. Journal of Management Information Systems, 9 (4), 37-57.
- Kiecker, P., Palan, K. & Areni, C. (2000). *Different Ways of "Seeing": How Gender Differences in Information Processing Influence the Content Analysis of Narrative Texts*. Marketing Letters, 11 (1), 49-65.
- Leech, S. & Sangster, A. (2002). *Expert systems in accounting research: a design science perspective*. In Arnold, V. & Sutton, S. (Eds.), *Researching accounting as an information systems discipline*, (pp. 65-80). Sarasota FL: American Accounting Association.
- Lewis, D. (1991). *Evaluating Text Categorization*. (University of Massachusetts) Amherst MA: Computer and Information Science Dept.
- Lewis, D. & Ringuette, M. (1994). *A Comparison of Two Learning Algorithms for Text Categorisation*. (April, 1994) Las Vegas: Symposium on Document Analysis and IR.
- Lombard, M., Snyder-Duch, J. & Campanella Bracken, C. (2004). *A Call for Standardization in Content Analysis Reliability*. Human Communication Research, 30 (3), 434-437.
- Mostafa, M., Quiroga, L. & Palakai, M. (1998). *Filtering Medical Documents Using Automated and Human Classification Methods*. Journal of the American Society for Information Science, 49 (14), 1304-1318.
- Nastase, V., Koeszegi, S. & Szpakowicz, S. (2007). *Content analysis through the machine learning mill*. Group Decision and Negotiation, 16 ((2007)), 335:346.
- Sebastiani, F. (1999). *A Tutorial on Automated Text Categorization*. (Consiglio Nazionale delle Ricerche) Pisa: Istituto di Elaborazione dell'Informazione.
- Sebastiani, F. (2002). *Machine learning in automated text categorization*. ACM Computing Surveys, 34 (1), 1-47.
- Snowball. (2008). *The Lovins stemming algorithm*. Retrieved 07/06/2008, from [snowball.tartarus.org/algorithms/lovins/stemmer.html](http://snowball.tartarus.org/algorithms/lovins/stemmer.html)

Subramani, M. & Walden, E. (2001). *The impact of E-commerce announcements on the Market Value of Firms*. Information Systems Research, 12 (2), 135-154.

Working group on Libre software. (2000). *Free software / Open source: Information Society Opportunities for Europe?* (Version 1.2) Brussels: Information Society Directorate General of the European Commission.

[references to suppliers removed]

[references to author removed]

## **Appendix A: Selection of Software for Automated Text Categorisation**

In this appendix, I report how I have selected a software package for automated text categorisation. The approach used was the following supplier selection process:

- Create a longlist of 10 to 20 potential suppliers and products based on market research
- Reduce the longlist to a shortlist of 2 to 4 potential suppliers and products based on supplier and product review
- Reduce the shortlist to a preliminary selection of 2 suppliers and products; one is the preferred supplier and the other is the backup in case commercial terms cannot be agreed

The remainder of this appendix has been taken out of this version of the paper, as results have not yet been shared with the suppliers of the software.

## Tables

**Table 1 Effectiveness measures for automated text categorisation**

Measure Name	Calculation Formula
Recall	$(TP) / (TP + FN)$
Precision	$(TP) / (TP + FP)$
Error	$(FN + FP) / (TP + FP + FN + TN)$
Accuracy	1 - error
precision-recall break-even point	the value at which precision and recall are equal
fall-out	$(FP) / (FP + TN)$
Overlap	$(TP) / (TP + FP + FN)$
$F_{\beta}$	$(\beta^2 + 1)(\text{precision} \times \text{recall}) / (\beta^2 \times \text{precision} + \text{recall})$ , a weighted average of precision and recall
utility functions	other weighted combinations of quality measures

**Table 2 Time requirement for manual categorisation of press releases**

Panel A: Actual time spent on manual categorisation of the AEX press release dataset

Step <sup>a</sup>	Input	Output	Actual time spent			
			hours	hours / 1,000 press releases	full time equivalent	time span (weeks) <sup>b</sup>
Researcher selects documents	List of AEX listed companies Factiva query	66,176 unclassified press releases in English	limited	limited	limited	limited
Researcher trains coders	Coding guideline 500 unclassified press releases	500 press releases classified by two coders Revised coding guideline	limited	limited	limited	limited
Coders categorise documents	Revised coding guideline 66,176 uncategorised English press releases	66,176 English press releases categorised in two classes: Yes (n=445 or 0.67%) No (n=65,731 or 99.3%)	397	6.00	0.29	36

Panel B: Prediction of time to be spent on manual categorisation of the LSE press release dataset

Step <sup>a</sup>	Input	Output	Predicted time to be spent			
			hours	hours / 1,000 press releases	full time equivalent	time span (weeks) <sup>b</sup>
Researcher selects documents	List of LSE listed companies Factiva query	342,992 unclassified press releases in English	limited	limited	limited	limited
Researcher trains coders	Coding guideline 500 unclassified press releases	500 press releases classified by several coders Revised coding guideline	limited	limited	limited	limited
Coders categorise documents	Revised coding guideline 342,992 uncategorised English press releases	342,992 English press releases categorised in two classes	6.00	2058	1.29	160

Note: Adapted from [reference to author]

<sup>a</sup> For an explanation of the steps, refer to Figure 1

<sup>b</sup> Coding was not done on a full time basis

**Table 3 Test and training datasets**

Name	Training documents				Test documents			
	Yes	No	Total	%	Yes	No	Total	%
left-right	223	32,866	33,089	50	222	32,865	33,087	50
major-minor	356	52,585	52,941	80	89	13,146	13,235	20
cod1-cod2	373	47,523	47,896	72	72	18,208	18,280	28

**Table 4 Model selection - Indexing for Package1 and Package2**

Parameter	Meaning	Values tested	Meaning
Stemming	Stemming is used when it can be assumed that different word forms based on the same stem are equivalent with respect to the classification task. Stemming algorithms reduce words to stem terms	Porter	Simple rule based algorithm for stemming English words [Joachims 2001]
		Lovins	Algorithm designed by Lovins in 1968 [Snowball 2008]
		ToLowerCase	Algorithm that converts stems to lower case letters [supplier reference removed]
Wordfiltering	Wordfiltering is used when certain words can be assumed not to be relevant for the classification task at hand. Wordfiltering excludes terms that occur frequently in any text, like definite and indefinite articles.	EnglishStopWords	Standard stopwords list for the English language[supplier reference removed]
Termweighting	In order to determine the importance of terms in the input, each term gets a weight. Weights can depend on the number of occurrences of a term in a document, the number of occurrences of a term in all documents, or the total number of terms in all documents.	BinaryOccurrences	Each term in a document has weight 1, all terms that do not occur in the same document have weight 0 [Joachims 2001]
		TermFrequency	Each term in a document gets the number of its occurrences divided by the total number of terms in this document as a weight [supplier reference removed]
		TFIDF	Each term gets its term frequency weighed again by the total number of terms in all documents as a weight [supplier reference removed]
		TermOccurences	Each term in a document gets the number of its occurrences as a weight [Joachims 2001]
Minchars	The minimum number of characters a term must have to be processed in the classification		2
			3
			4
			6
			8
		10	

**Table 5 Model selection - Stopwords for Package1 and Package2**

Stopwords	
sap	qad
oracle	oneworld
peoplesoft	navision
edwards	movex
sage	jba
microsoft	baan
intenia	mfg
geac	mysap
invensys	erp
ssa	

**Table 6 Linear regression variables**

Variable	Description
F	$F_{\beta=4}$ , the F-value of a text categorisation run (%)
SECONDS	The total training and test time for a text categorisation run (seconds)
C	Constant term in the regression
CORPUS_LR	Dummy value, 1 if the run was carried out on the test and training set leftright, 0 otherwise
CORPUS_MM	Dummy value, 1 if the run was carried out on the test and training set majorminor, 0 otherwise
STEMMER_LV	Dummy value, 1 if the datasets were stemmed with the Lovins stemmer, 0 otherwise
STEMMER_PT	Dummy value, 1 if the datasets were stemmed with the Porter stemmer, 0 otherwise
TERMW_BO	Dummy value, 1 if terms in the datasets were weighted with Binary Occurrences, 0 otherwise
TERMW_TF	Dummy value, 1 if terms in the datasets were weighted with Term Frequency, 0 otherwise
TERMW_TO	Dummy value, 1 if terms in the datasets were weighted with Term Occurrences, 0 otherwise
MINCHAR	The minimum number of characters a term must have to be processed in the classification

**Table 7 Package1 and Package2 categorisation results**

Corpus	Stemmer	Stopwords	Term Weight	Min char	Package1		F-value ( $\beta=4$ )	Package2		F-value ( $\beta=4$ )	
					Precision (%)	Recall (%)		Precision (%)	Recall (%)		
leftright	Lovins	English	Binary Occurences	2	86.7	11.7	12.3	0.4	20.3	5.2	
				3	86.7	11.7	12.3	0.5	45.9	7.2	
				4	86.2	11.3	11.9	0.7	68.5	10.2	
				6	85.7	10.8	11.4	0.7	71.6	10.3	
				8	84.6	9.9	10.4	0.6	29.7	7.7	
				10	66.7	7.2	7.6	0.7	14.0	6.6	
				Term Frequency	2	80.0	9.0	9.5	1.1	54.1	14.1
					3	80.8	9.5	10.0	0.7	44.6	9.5
					4	77.8	9.5	10.0	0.7	68.5	10.2
					6	79.2	8.6	9.1	0.7	92.8	10.6
			Term Occurrences	8	60.0	9.5	10.0	0.7	74.3	10.3	
				10	100.0	0.5	0.5	0.6	87.8	9.2	
				2	72.7	3.6	3.8	0.8	35.6	10.0	
				3	61.5	3.6	3.8	0.6	39.2	8.2	
				4	61.5	3.6	3.8	0.8	44.1	10.5	
				6	61.5	3.6	3.8	0.6	59.5	8.8	
				8	70.0	6.3	6.7	0.6	63.5	8.9	
				10	100.0	0.9	1.0	0.4	17.6	5.0	
				TFIDF	2	80.8	28.4	29.5	0.8	95.0	12.0
					3	77.5	27.9	29.0	0.7	90.1	10.6
			4		76.5	23.4	24.4	0.7	96.4	10.7	
			6		76.0	17.1	17.9	0.7	97.7	10.7	
			8		63.0	15.3	16.0	0.6	86.0	9.2	
			10		88.9	3.6	3.8	0.6	65.3	8.9	
	Porter	English	Binary Occurences		2	85.7	10.8	11.4	0.3	26.6	4.3
					3	85.7	10.8	11.4	0.5	51.8	7.4
				4	85.7	10.8	11.4	0.7	53.2	9.8	
				6	85.7	10.8	11.4	0.6	64.0	8.9	
				8	84.0	9.5	10.0	0.4	18.9	5.1	
				10	63.2	5.4	5.7	1.0	23.9	10.2	
			Term Frequency	2	80.8	9.5	10.0	0.5	45.5	7.2	
				3	80.0	9.0	9.5	0.6	73.9	9.0	
				4	81.5	9.9	10.4	0.6	57.2	8.7	
				6	77.8	9.5	10.0	0.7	86.5	10.5	
				8	59.5	9.9	10.4	0.5	33.3	6.9	
				10	100.0	0.5	0.5	0.7	97.7	10.7	
		Term Occurrences	2	66.7	3.6	3.8	0.5	46.4	7.3		
			3	66.7	3.6	3.8	0.7	71.2	10.3		
			4	61.5	3.6	3.8	0.6	33.8	7.9		
			6	72.7	3.6	3.8	0.5	44.6	7.2		
			8	70.0	6.3	6.7	0.4	22.1	5.3		
			10	100.0	0.9	1.0	0.5	25.2	6.5		
			TFIDF	2	76.8	28.4	29.5	0.7	95.5	10.7	
				3	72.4	28.4	29.5	0.7	98.6	10.7	
				4	70.6	27.0	28.0	0.7	95.0	10.6	
				6	64.9	16.7	17.5	0.7	96.8	10.7	
				8	57.6	15.3	16.0	0.6	76.6	9.1	
				10	87.5	3.2	3.4	0.7	80.6	10.4	

Corpus	Stemmer	Stopwords	Term Weight	Min char	Package1			Package2		
					Precision (%)	Recall (%)	F-value ( $\beta=4$ )	Precision (%)	Recall (%)	F-value ( $\beta=4$ )
cod1cod2	ToLower	English	Binary Occurences	2	85.7	2.7	2.9	0.5	51.4	7.4
				3	100.0	1.8	1.9	0.6	54.5	8.7
				4	83.3	4.5	4.8	0.7	52.7	9.8
				6	100.0	2.3	2.4	0.6	68.9	9.0
				8	100.0	0.9	1.0	0.6	39.6	8.2
			10	100.0	0.5	0.5	0.5	18.0	5.9	
			Term Frequency	2	NaN	0.0	NaN	0.6	64.9	8.9
				3	NaN	0.0	NaN	0.7	100.0	10.7
				4	NaN	0.0	NaN	1.0	61.7	13.5
				6	NaN	0.0	NaN	0.6	75.2	9.0
				8	NaN	0.0	NaN	0.7	75.2	10.4
			10	NaN	0.0	NaN	0.6	77.5	9.1	
			Term Occurences	2	100.0	0.9	1.0	0.5	58.6	7.5
				3	66.7	0.9	1.0	0.5	49.5	7.3
				4	63.6	3.2	3.4	0.7	42.3	9.4
				6	70.0	3.2	3.4	0.6	54.5	8.7
				8	100.0	0.9	1.0	0.5	39.6	7.1
			10	NaN	0.0	NaN	0.4	19.8	5.1	
			TFIDF	2	100.0	0.5	0.5	0.7	98.6	10.7
					3	100.0	0.5	0.5	0.7	99.1
	4	NaN			0.0	NaN	0.7	100.0	10.7	
	6	NaN			0.0	NaN	0.7	96.4	10.7	
	8	NaN			0.0	NaN	0.6	90.1	9.2	
	10	NaN		0.0	NaN	0.5	68.0	7.6		
		14.3		2.8	2.9	1.1	47.2	13.6		
		16.7		2.8	2.9	0.5	8.3	4.3		
		15.4		2.8	2.9	0.4	8.3	3.8		
		27.3		4.2	4.4	1.1	47.2	13.6		
	28.6	5.6	5.9	0.3	6.9	3.0				
	11.1	1.4	1.5	0.7	38.9	9.2				
	Term Frequency	2	12.5	1.4	1.5	NaN	0.0	NaN		
		3	12.5	1.4	1.5	0.9	1.4	1.4		
		4	25.0	2.8	3.0	1.3	9.7	7.0		
		6	NaN	0.0	NaN	1.4	37.5	14.9		
		8	NaN	0.0	NaN	0.1	1.4	0.8		
	10	NaN	0.0	NaN	0.7	90.3	10.6			
	Term Occurences	2	11.1	1.4	1.5	0.3	22.2	4.2		
		3	10.0	1.4	1.5	0.2	8.3	2.5		
		4	11.1	1.4	1.5	0.3	20.8	4.1		
		6	11.1	1.4	1.5	0.6	75.0	9.0		
8		8.3	1.4	1.5	0.2	19.4	2.9			
10	NaN	0.0	NaN	0.6	55.6	8.7				
TFIDF	2	14.8	5.6	5.8	0.4	26.4	5.5			
		12.5	5.6	5.8	0.3	15.3	3.9			
		20.0	6.9	7.2	0.4	20.8	5.2			
		14.3	2.8	2.9	0.8	59.7	11.2			
		7.7	1.4	1.5	0.2	15.3	2.8			
	10	NaN	0.0	NaN	0.6	66.7	8.9			
		18.2	2.8	2.9	0.1	8.3	1.4			
		18.2	2.8	2.9	0.1	2.8	1.1			
		18.2	2.8	2.9	0.4	8.3	3.8			
		18.2	2.8	2.9	0.1	8.3	1.4			

Corpus	Stemmer	Stopwords	Term Weight	Min char	Package1			Package2			
					Precision (%)	Recall (%)	F-value ( $\beta=4$ )	Precision (%)	Recall (%)	F-value ( $\beta=4$ )	
				6	20.0	2.8	2.9	0.3	6.9	3.0	
				8	18.2	2.8	2.9	0.1	8.3	1.4	
				10	18.2	2.8	2.9	0.1	5.6	1.3	
				2	22.2	2.8	3.0	0.3	6.9	3.0	
				Frequency	3	NaN	0.0	NaN	2.1	8.3	7.1
					4	22.2	2.8	3.0	NaN	0.0	NaN
				Term	6	NaN	0.0	NaN	0.4	2.8	2.1
					8	NaN	0.0	NaN	0.1	1.4	0.8
				Occurrences	10	NaN	0.0	NaN	0.2	11.1	2.6
					2	11.1	1.4	1.5	0.2	20.8	2.9
				TFIDF	3	8.3	1.4	1.5	0.3	26.4	4.3
					4	11.1	1.4	1.5	0.1	5.6	1.3
				Term	6	11.1	1.4	1.5	0.1	2.8	1.1
					8	11.1	1.4	1.5	0.2	25.0	3.0
				Occurrences	10	NaN	0.0	NaN	0.1	6.9	1.4
					2	15.6	6.9	7.1	0.3	20.8	4.1
				TFIDF	3	17.2	6.9	7.2	0.5	16.7	5.7
					4	20.0	6.9	7.2	0.6	19.4	6.8
				Term	6	14.3	2.8	2.9	0.2	16.7	2.9
					8	7.7	1.4	1.5	0.7	25.0	8.2
				Occurrences	10	NaN	0.0	NaN	0.2	8.3	2.5
					2	46.2	8.3	8.7	0.4	25.0	5.4
				TFIDF	3	38.5	6.9	7.3	0.7	20.8	7.7
					4	33.3	5.6	5.9	0.5	56.9	7.5
				Term	6	41.7	6.9	7.3	0.3	16.7	4.0
					8	41.7	6.9	7.3	0.1	6.9	1.4
				Occurrences	10	33.3	4.2	4.4	0.8	33.3	9.8
					2	14.3	1.4	1.5	NaN	0.0	NaN
				TFIDF	3	NaN	0.0	NaN	NaN	0.0	NaN
					4	22.2	2.8	3.0	1.1	11.1	7.2
				Term	6	NaN	0.0	NaN	NaN	0.0	NaN
					8	NaN	0.0	NaN	0.1	1.4	0.8
				Occurrences	10	NaN	0.0	NaN	0.7	56.9	9.9
					2	10.0	1.4	1.5	0.5	37.5	7.0
				TFIDF	3	10.0	1.4	1.5	0.5	37.5	7.0
					4	11.1	1.4	1.5	0.5	61.1	7.5
				Term	6	8.3	1.4	1.5	0.1	13.9	1.5
					8	9.1	1.4	1.5	0.1	11.1	1.5
				Occurrences	10	NaN	0.0	NaN	0.6	47.2	8.5
					2	22.2	5.6	5.9	0.8	37.5	10.1
TFIDF	3	22.2	5.6	5.9	0.8	40.3	10.3				
	4	13.6	4.2	4.4	0.6	52.8	8.6				
Term	6	11.8	2.8	2.9	0.3	26.4	4.3				
	8	NaN	0.0	NaN	0.1	9.7	1.5				
Occurrences	10	NaN	0.0	NaN	0.8	61.1	11.2				
	2	75.0	16.9	17.7	0.6	57.3	8.7				
TFIDF	3	78.9	16.9	17.7	0.7	62.9	10.1				
	4	82.4	15.7	16.5	1.3	58.4	16.3				
Term	6	88.9	18.0	18.9	0.7	47.2	9.6				
	8	82.4	15.7	16.5	0.6	20.2	6.9				
Occurrences	10	73.7	15.7	16.5	0.6	15.7	6.3				

Corpus	Stemmer	Stopwords	Term Weight	Min char	Package1			Package2		
					Precision (%)	Recall (%)	F-value ( $\beta=4$ )	Precision (%)	Recall (%)	F-value ( $\beta=4$ )
			Term	2	86.7	14.6	15.4	0.6	7.9	4.6
			Frequency	3	85.7	13.5	14.2	0.4	12.4	4.5
				4	90.0	10.1	10.7	1.4	31.5	13.9
				6	100.0	6.7	7.1	1.0	37.1	11.9
				8	90.0	10.1	10.7	0.8	46.1	10.6
				10	NaN	0.0	NaN	0.8	91.0	11.9
			Term Occurrences	2	62.5	5.6	5.9	0.4	23.6	5.3
				3	70.0	7.9	8.3	0.5	30.3	6.7
				4	70.0	7.9	8.3	0.5	32.6	6.8
				6	70.0	7.9	8.3	0.5	25.8	6.5
				8	70.0	7.9	8.3	0.7	20.2	7.7
				10	100.0	2.2	2.3	1.0	44.9	12.5
			TFIDF	2	79.1	38.2	39.4	0.7	66.3	10.2
				3	78.0	36.0	37.2	0.6	61.8	8.8
				4	90.0	30.3	31.5	0.8	64.0	11.3
				6	91.7	24.7	25.8	0.7	68.5	10.2
				8	80.0	18.0	18.9	0.8	71.9	11.5
				10	100.0	5.6	5.9	0.7	62.9	10.1
	Porter	English	Binary Occurrences	2	81.0	19.1	20.0	0.7	50.6	9.7
				3	80.0	18.0	18.9	0.8	49.4	10.8
				4	75.0	16.9	17.7	0.6	37.1	8.1
				6	78.9	16.9	17.7	0.5	32.6	6.8
				8	77.8	15.7	16.5	0.6	14.6	6.2
				10	72.2	14.6	15.3	0.4	12.4	4.5
			Term Frequency	2	86.7	14.6	15.4	0.4	7.9	3.8
				3	85.7	13.5	14.2	1.0	18.0	9.0
				4	90.9	11.2	11.8	0.2	9.0	2.5
				6	100.0	6.7	7.1	1.1	34.8	12.4
				8	72.7	9.0	9.5	1.0	38.2	12.0
				10	NaN	0.0	NaN	0.7	89.9	10.6
			Term Occurrences	2	62.5	5.6	5.9	0.8	39.3	10.3
				3	70.0	7.9	8.3	1.0	47.2	12.7
				4	70.0	7.9	8.3	0.5	27.0	6.6
				6	70.0	7.9	8.3	0.6	15.7	6.3
				8	70.0	7.9	8.3	0.1	5.6	1.3
				10	100.0	2.2	2.3	0.5	22.5	6.3
			TFIDF	2	76.9	33.7	34.9	0.7	68.5	10.2
				3	75.7	31.5	32.6	0.7	74.2	10.3
				4	81.3	29.2	30.3	0.6	69.7	9.0
				6	84.6	24.7	25.8	0.7	60.7	10.0
				8	78.3	20.2	21.1	0.8	70.8	11.5
				10	100.0	5.6	5.9	0.5	50.6	7.3
	ToLower	English	Binary Occurrences	2	81.3	14.6	15.3	0.8	65.2	11.4
				3	75.0	13.5	14.2	0.5	49.4	7.3
				4	76.5	14.6	15.3	0.8	74.2	11.6
				6	80.0	13.5	14.2	0.7	70.8	10.3
				8	78.6	12.4	13.0	0.5	23.6	6.3
				10	57.1	4.5	4.8	0.3	19.1	4.1
			Term Frequency	2	66.7	2.2	2.3	0.9	4.5	3.6
				3	66.7	2.2	2.3	0.4	3.4	2.4
				4	100.0	2.2	2.3	0.4	27.0	5.5

Corpus	Stemmer	Stopwords	Term Weight	Min char	Package1			Package2		
					Precision (%)	Recall (%)	F-value ( $\beta=4$ )	Precision (%)	Recall (%)	F-value ( $\beta=4$ )
				6	100.0	3.4	3.6	0.5	22.5	6.3
				8	100.0	3.4	3.6	0.8	40.4	10.3
				10	NaN	0.0	NaN	0.6	68.5	8.9
			Term Occurrences	2	40.0	2.2	2.3	0.6	34.8	8.0
				3	33.3	2.2	2.3	0.3	22.5	4.2
				4	60.0	6.7	7.1	0.8	64.0	11.3
				6	63.6	7.9	8.3	0.9	77.5	12.9
				8	57.1	4.5	4.8	0.6	23.6	7.3
			TFIDF	10	100.0	2.2	2.3	0.3	18.0	4.0
				2	50.0	3.4	3.6	0.8	87.6	11.9
				3	50.0	3.4	3.6	0.7	73.0	10.3
				4	57.1	4.5	4.8	0.7	85.4	10.5
				6	60.0	6.7	7.1	0.7	89.9	10.6
				8	60.0	6.7	7.1	0.6	69.7	9.0
				10	100.0	1.1	1.2	0.6	59.6	8.8
Average over valid runs					60.7	8.2	8.6	0.6	43.8	7.7

**Table 8 Package3 results**

Case	Precision (%)	Recall (%)	F-value ( $\beta=4$ )
leftright	4.0	90.1	39.8
cod1cod2	4.9	90.3	44.6
majorminor	4.5	91.0	42.7
Average	4.5	90.5	42.4

**Table 9 Package1 timing results**

Corpus	Stemmer	Term Weight	Min char	F-value ( $\beta=4$ )	Time (seconds)
majorminor	PorterStemmer	BinaryOccurrences	2	20.0	1,902.6
			3	18.8	1,680.7
			4	17.7	1,739.7
			6	17.7	1,620.0
			8	16.5	929.4
		10	15.3	410.8	
		TermFrequency	2	15.4	1,797.3
			3	14.2	1,965.3
			4	11.8	1,663.3
			6	7.1	1,155.4
			8	9.5	790.0
		TFIDF	2	34.9	1,993.7
			3	32.6	1,620.0
			4	30.4	1,808.3
			6	25.8	1,510.7
	8		21.1	857.4	
	10	5.9	365.8		
	TermOccurrences	2	5.9	1,586.1	
		3	8.3	1,765.7	
		4	8.3	1,834.6	
		6	8.3	1,322.2	
		8	8.3	611.1	
	10	2.4	351.6		
	LovinsStemmer	BinaryOccurrences	2	17.7	1,755.3
			3	17.7	1,450.0
			4	16.5	1,288.4
			6	18.9	1,142.9
			8	16.5	651.1
		10	16.5	405.5	
		TermFrequency	2	15.4	1,228.2
			3	14.2	1,389.5
			4	10.7	1,370.0
			6	7.1	1,055.8
			8	10.7	939.5
		TFIDF	2	39.4	2,888.7
			3	37.1	1,604.9
			4	31.6	1,053.0
			6	25.8	1,263.1
	8		18.8	835.2	
	10	5.9	397.6		
	TermOccurrences	2	5.9	1,421.3	
		3	8.3	1,382.0	
		4	8.3	1,089.4	
		6	8.3	1,237.7	
		8	8.3	1,000.0	
10	2.4	357.1			
ToLowerCaseConverter	BinaryOccurrences	2	15.3	2,414.5	
		3	14.2	2,089.4	
		4	15.3	1,880.6	

Corpus	Stemmer	Term Weight	Min char	F-value ( $\beta=4$ )	Time (seconds)		
	Stemmer		6	14.2	1,453.5		
			8	13.0	907.7		
			10	4.8	531.7		
			TermFrequency	2	2.4	2,129.8	
				3	2.4	1,488.3	
				4	2.4	1,790.0	
			TFIDF	6	3.6	2,046.4	
				8	3.6	916.6	
				2	3.6	1,751.9	
				3	3.6	2,051.7	
		TermOccurrences	4	4.8	2,349.7		
			6	7.1	1,870.8		
			8	7.1	874.1		
			10	1.2	495.1		
			2	2.4	1,979.0		
			3	2.4	1,630.2		
			PorterStemmer	BinaryOccurrences	4	7.1	1,375.1
					6	8.3	1,625.5
					8	4.8	709.8
					10	2.4	319.0
2	11.4				1,404.3		
3	11.4				1,338.5		
4	11.4				1,160.7		
6	11.4				1,535.8		
8	10.0				729.1		
10	5.7				308.6		
	PorterStemmer	TFIDF	2	10.0	1,303.9		
			3	9.5	1,319.9		
			4	10.4	1,633.6		
			6	10.0	1,408.4		
			8	10.4	698.5		
			10	0.5	292.5		
			2	29.5	1,542.8		
			3	29.4	1,860.8		
			4	28.0	1,126.8		
			6	17.4	1,076.0		
	PorterStemmer	TermOccurrences	8	16.0	664.7		
			10	3.3	339.0		
			2	3.8	1,393.3		
			3	3.8	1,842.2		
			4	3.8	1,288.9		
			6	3.8	1,331.1		
			8	6.7	676.5		
			10	1.0	326.1		
			LovinsStemmer	BinaryOccurrences	2	12.3	1,555.0
					3	12.3	1,155.8
4	11.9	1,128.6					
6	11.4	1,324.2					
8	10.5	675.5					
10	7.6	271.1					
	LovinsStemmer	TermFrequency	2	9.5	1,201.0		
			3	10.0	1,012.0		

Corpus	Stemmer	Term Weight	Min char	F-value ( $\beta=4$ )	Time (seconds)	
		TFIDF	4	10.0	1,095.7	
			6	9.0	864.4	
			8	10.0	684.3	
			10	0.5	295.2	
			2	29.5	1,591.3	
			3	29.0	1,293.0	
			4	24.4	1,068.9	
			6	17.9	1,051.8	
			8	16.0	680.0	
			10	3.8	430.1	
		TermOccurrences	2	3.8	1,802.5	
			3	3.8	1,472.4	
			4	3.8	1,299.3	
			6	3.8	1,298.3	
			8	6.7	786.0	
			10	1.0	314.8	
			ToLowerCaseConverter BinaryOccurrences	2	2.9	1,419.4
				3	1.9	1,746.3
				4	4.8	1,612.6
				6	2.4	1,054.8
		8		1.0	880.4	
		10		0.5	426.1	
		TFIDF		2	0.5	1,176.8
				3	0.5	1,554.4
				TermOccurrences	2	1.0
		3			1.0	1,860.8
		4	3.3		1,472.0	
		cod1cod2 PorterStemmer BinaryOccurrences	6	3.3	1,410.8	
			8	1.0	976.3	
			2	2.9	1,983.4	
3	2.9		1,569.5			
4	2.9		2,152.2			
6	2.9		1,864.9			
8	2.9		859.4			
10	2.9		455.3			
TermFrequency	2		2.9	1,813.0		
	4		2.9	1,864.9		
TFIDF	2	7.2	1,642.0			
	3	7.2	2,262.2			
	4	7.2	2,005.9			
	6	2.9	1,690.0			
	8	1.5	1,011.7			
	TermOccurrences	2	1.5	2,274.3		
3		1.5	1,782.9			
4		1.5	1,722.7			
6		1.5	1,619.9			
8		1.5	1,067.8			
cod1cod2 LovinsStemmer BinaryOccurrences		2	2.9	1,817.1		
	3	2.9	1,387.7			
	4	2.9	1,534.5			
	6	4.4	1,283.6			
	8	5.8	620.5			

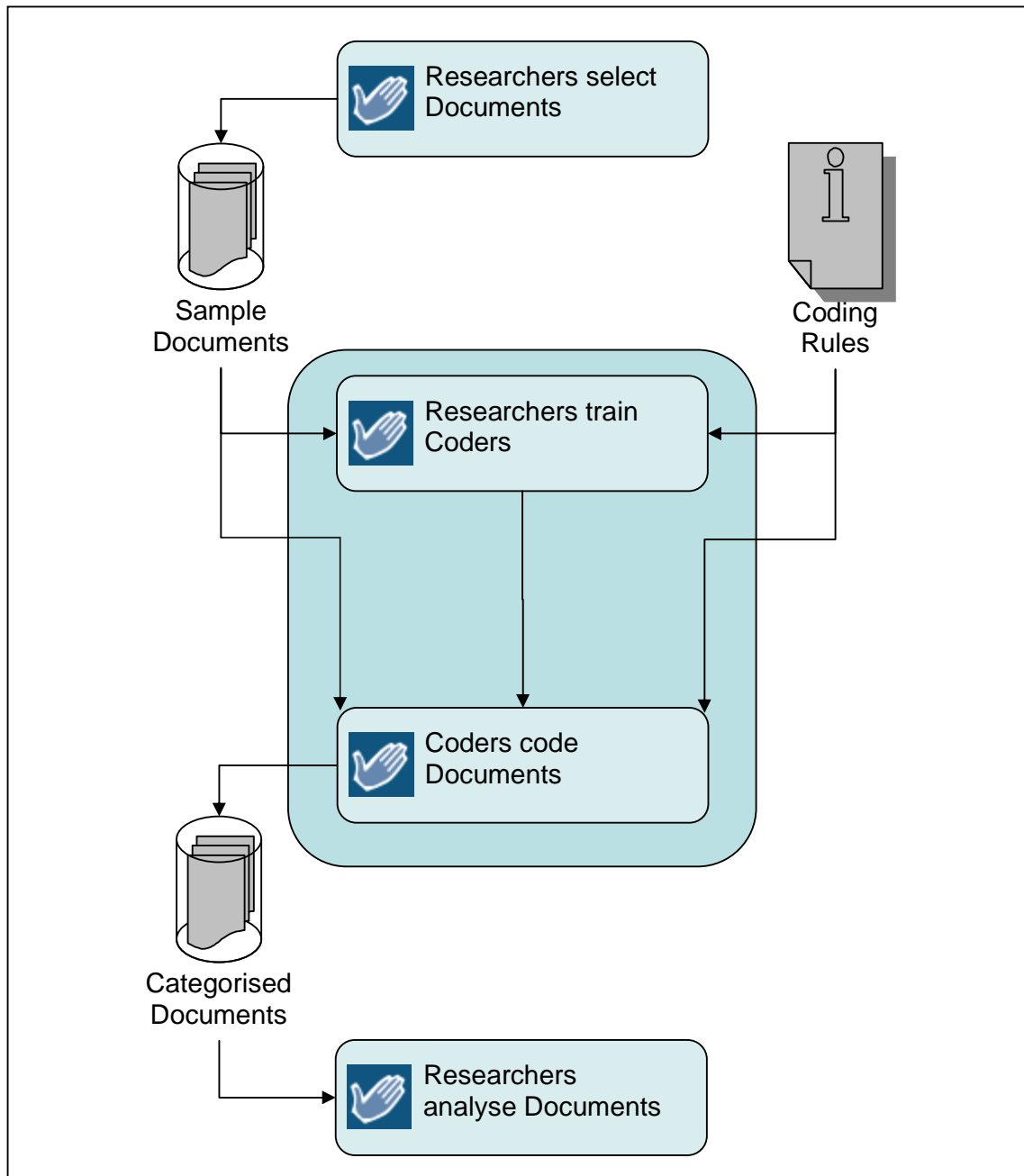
Corpus	Stemmer	Term Weight	Min char	F-value ( $\beta=4$ )	Time (seconds)
			10	1.5	290.8
		TermFrequency	2	1.5	1,418.0
			3	1.5	1,951.5
			4	2.9	1,170.0
		TFIDF	2	5.8	1,536.2
			3	5.7	1,322.5
			4	7.2	1,422.3
			6	2.9	1,309.0
			8	1.5	595.2
		TermOccurrences	2	1.5	3,114.2
			3	1.5	1,524.6
			4	1.5	1,434.8
			6	1.5	1,347.8
			8	1.5	968.0
	ToLowerCaseConverter	BinaryOccurrences	2	8.8	3,617.7
			3	7.3	2,200.4
			4	5.8	2,117.8
			6	7.3	1,869.1
			8	7.3	1,070.3
			10	4.4	509.4
		TermFrequency	2	1.5	2,131.2
			4	2.9	2,470.9
		TFIDF	2	5.8	1,975.8
			3	5.8	2,020.6
			4	4.3	1,718.2
			6	2.9	1,771.5
		TermOccurrences	2	1.5	1,561.0
			3	1.5	2,290.1
			4	1.5	2,305.8
			6	1.5	1,886.8
			8	1.5	1,159.7
Average				8.6	1,359.8

**Table 10 Regression results**

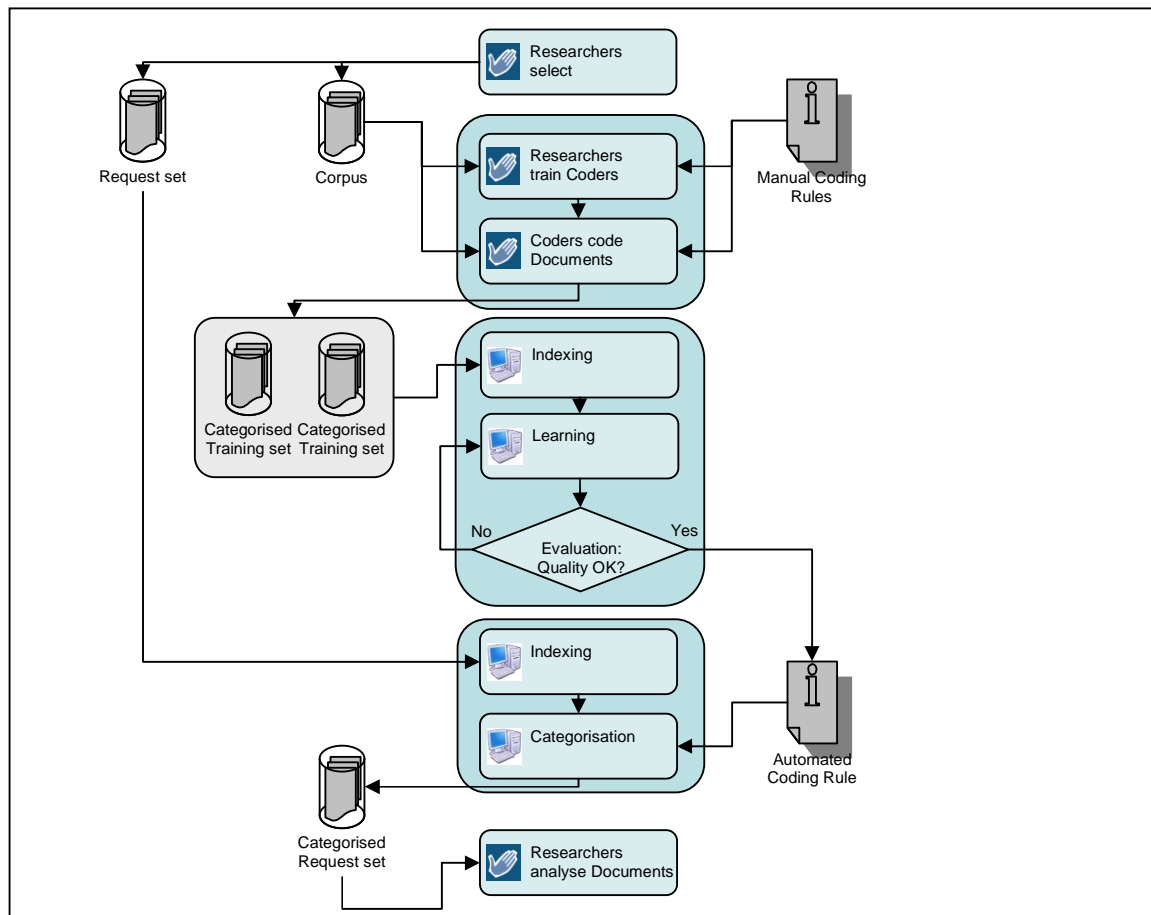
Regression model	M2a		M2b				M2c	
Dependent variable	F		SECONDS				F	
Independent variables	$\alpha$	$t(\alpha)$	$\gamma$	$t(\gamma)$	$\delta$	$t(\delta)$		
C	8.78	7.35 (**)	2435.73	31.02 (**)	7.25	4.76 (**)		
CORPUS_LR	4.97	5.18 (**)	-371.34	-5.88 (**)				
CORPUS_MM	7.68	7.99 (**)	-133.10	-2.11 (**)				
STEMMER_LV	11.62	8.08 (**)	-165.15	-1.75 (*)				
STEMMER_PT	4.28	5.33 (**)	2.51	0.05				
TERMW_BO	-2.76	-2.70 (**)	8.85	0.13				
TERMW_TF	-5.67	-4.81 (**)	-93.94	-1.21				
TERMW_TO	-7.59	-7.07 (**)	3.28	0.05				
MINCHAR	-0.65	-4.60 (**)	-171.04	-18.36 (**)				
SECONDS					0.00	0.99		

## Figures

Figure 1 Manual text categorisation process



**Figure 2 Automated text categorisation process with machine learning**



**Figure 3** Template for a contingency table

		Manual categorisation	
		Yes	No
Learner categorisation	Yes	True Positive (TP)	False Positive (FP)
	No	False Negative (FN)	True Negative (TN)

**Figure 4 Precision versus Recall for Package1, Package2 and Package3**

