

Labeling Clinical Reports with Active Learning and Topic Model- ing

*Uppmärkning av kliniska rapporter genom active learning och
topic modeller*

Simon Lindblad

Supervisor : Marco Kuhlmann
Examiner : Arne Jönsson

External supervisor : Mikael Nilsson

Upphovsrätt

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår. Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art. Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart. För ytterligare information om Linköping University Electronic Press se förlagets hemsida <http://www.ep.liu.se/>.

Copyright

The publishers will keep this document online on the Internet – or its possible replacement – for a period of 25 years starting from the date of publication barring exceptional circumstances. The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility. According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement. For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: <http://www.ep.liu.se/>.

Abstract

Supervised machine learning models require a high quality set of labeled data in order to perform well. This set of data is sometimes easy to obtain, but when it comes to text that is usually not the case. Available text data often exists in abundance, but it is seldom labeled. Labeling text data is a time consuming process, especially in the case where multiple labels can be assigned to a single text document.

The purpose of this thesis work was to make the labeling process of clinical reports as effective and effortless as possible by evaluating different multi-label active learning techniques. By using one of these techniques the goal was to reduce the number of labeled documents needed, and increase the quality of those that is. Binary Version Space Minimization, Maximum Loss Reduction with Maximum Confidence and Adaptive Active Learning are the three methods evaluated.

The Reuters dataset was used for comparisons and simulation of the labeling process. Using Binary Version Space Minimization, an accuracy of 89% was achieved with 2500 reports, compared to 85% with random sampling. With Adaptive Active Learning, 85% accuracy could be reached after labeling 975 reports, compared to 1700 reports with random sampling.

The focus was on clinical reports, and a part of this study was to filter out the reports describing examinations that did not occur. This was done by using an LDA model, where a couple of topics were able to identify reports describing canceled examinations.

Acknowledgments

Acknowledgments.tex

Contents

Abstract	iii
Acknowledgments	iv
Contents	v
List of Figures	vii
List of Tables	ix
1 Introduction	1
1.1 Motivation	1
1.2 Aim	3
1.3 Research questions	4
1.4 Delimitations	5
1.5 Structure of the Report	5
2 Theory	6
2.1 Text Classification	6
2.1.1 Support Vector Machines	7
2.1.2 Multi-Label Classification	8
2.2 Active Learning	8
2.2.1 Pool-Based Sampling	9
2.2.2 The Version Space	9
2.2.3 Binary Version Space Minimization	10
2.2.4 Maximum Loss Reduction with Maximum Confidence	10
2.2.5 Adaptive Active Learning	12
2.3 Text Processing using Unsupervised Techniques	13
2.3.1 Topic Modeling	14
2.3.2 Text Clustering	16
2.3.3 Word Embeddings	17
2.4 Evaluation Metrics	18
3 Data	20
3.1 Datasets	20
3.1.1 Clinical Dataset From Sectra	20
3.1.2 Reuters-21578	22
3.2 Pre-Processing and Text Representation	23
3.3 Exploratory Study	26
3.3.1 The Relationship Between Topics and Categories	27
3.3.2 The Relationship Between Topics and Invalid Reports	29
4 Experiments	33

4.1	Experiment 1: How Well do the Sampling Strategies Perform?	33
4.1.1	Method	33
4.1.2	Results	35
4.2	Experiment 2: Evaluating the Label Balance on the Labeled Dataset	40
4.2.1	Method	40
4.2.2	Results	41
4.3	Experiment 3: Filtering out Invalid Reports	44
4.3.1	Method	44
4.3.2	Results	45
4.4	Frameworks, Tools and Implementation	45
5	Discussion	47
5.1	Results	47
5.2	Method	49
5.3	Related Work	50
5.4	The Work in a Wider Context	51
6	Conclusion	52
	Bibliography	54

List of Figures

1.1	An overview of an active learning system. The dotted lines represents the things that some samplings strategies base their calculations on.	2
1.2	A screenshot of the interface used to label clinical reports at Sectra. A report is presented to the user, who can then select whichever categories (the blue buttons) deemed appropriate. By pressing “next”, the system presents a new reports.	3
2.1	Diagram of the LDA model.	15
2.2	(a) to (e) shows iterations of k -means until convergence. In (e) it can be seen that the new centroids capture the same documents as the previous iteration, and we have converged. The circles represents the seeds for the clusters and the data points are represented by squares. The color of the points are shows which cluster the point is currently assigned to.	18
3.1	A sample report from the dataset provided by Sectra	21
3.2	The distribution over the labels in the initial set of labeled data provided by Sectra	22
3.3	The distribution over the labels in the Reuters data	23
3.4	Two figures illustrating how the names were discovered in the word2vec plot. The (b) plot represents the red square in (a).	25
3.5	The perplexity scores for the different LDA models	26
3.6	A 2D plot of the text data, where each point is colored by topic with the highest probability	27
3.7	Wordclouds for a 75 topic LDA model	28
3.8	A way to visualize and analyze topics based on their relevance and frequency	28
3.9	The labeled data points plotted in 2D, and colored based on the first label of the report in alphabetical order.	29
3.10	The counts of the most likely topics for the four most common categories.	30
3.11	The categories of the different reports that are assigned a certain topic as the most likely one.	31
3.12	Distribution over the most likely topics for the valid and invalid reports. Note that only topics that occurred at least once are shown in the histogram.	31
3.13	The distribution of the number of prominent topics for the two categories.	32
4.1	Accuracy of the models with initial sample size 25	35
4.2	Accuracy of the models with initial sample size 50	36
4.3	Accuracy of the models with initial sample size 100	36
4.4	The metrics for the strategies when the initial sample size of 25 was used.	37
4.5	The metrics for the strategies when the initial sample size of 50 was used.	38
4.6	The metrics for the strategies when the initial sample size of 100 was used.	39
4.7	The percentage of time used on the different strategies during one iteration.	41
4.8	The distribution of labels after random sampling	42
4.9	The distribution of labels after BSVM	42
4.10	The distribution of labels after BSVM with clustering	42

4.11	The distribution of labels after MMC	43
4.12	The distribution of labels after Adaptive Active Learning	43
4.13	The distribution of labels after Adaptive Active Learning with clustering	43

List of Tables

2.1	Confusion matrix for explaining true positives, false positives, true negatives and false negatives	19
3.1	The synonyms, misspellings and shorts found in the data that the author could assert with confidence.	24
3.2	The number of reports assigned a certain category, as well as the number of different topics assigned as the most likely one for reports with the given category.	30
4.1	The different configurations of active learning strategies evaluated.	34
4.2	The number of labeled reports in total that the strategies required to achieve the different accuracy values, with initial sample size 25. Only results for the first 2500 data points that were labeled are considered.	40
4.3	The number of labeled reports in total that the strategies required to achieve the different accuracy values, with initial sample size 50. Only results for the first 2500 data points that were labeled are considered.	40
4.4	The number of labeled reports in total that the strategies required to achieve the different accuracy values, with initial sample size 100. Only results for the first 2500 data points that were labeled are considered.	40
4.5	The results after analyzing the label distribution after 500 new labels has been added.	44
4.6	The results after analyzing the label distribution after 2000 labels has been added.	44
4.7	The results of the classification of the invalid reports. The manual identification column represents the use of manual interpretation of the LDA topics to find the invalid reports.	45



1 Introduction

The world's population is growing each year. In order to raise to the challenges that arise with a growing population, it is of great importance that healthcare is made to be more efficient and robust. One way of increasing the efficiency as well as the quality of healthcare is to create automated systems that can aid doctors in their process. As the population grows, it is of utmost importance to ensure that the quality of diagnoses remain high, and that the risk of missing some critical piece of information is minimized. Taking advantage of the available medical information is key to create such systems.

Most of the approaches that exists today to create the aforementioned automated systems need a set of categorized data in order to identify and exploit certain patterns. The purpose of this thesis is to evaluate different techniques for labeling multi-label clinical reports. The goal is to make the set of labeled reports more useful in future systems by labeling them based on a strategy instead of selecting them at random. By using a selection strategy, the need for a large set of labeled reports could be reduced. The work is done at Sectra Medical Imaging IT Solutions AB in connection with a research project they have with Region Skåne. Region Skåne is responsible for the healthcare in Skåne, the southern most county of Sweden. This research project project on investigating how to use machine learning and text mining techniques to improve the functionality of their products and aid the physicians in their work.

1.1 Motivation

Information pertaining to a patient's diagnosis is often in the form of written clinical reports. This is a good example of data that can be utilized in automated systems. By extracting information from old reports, the process of writing new ones can be simplified. A system could show cases with similar features as the one currently being written, the doctor could then compare the findings and check if they have obtained an abnormal result. Being able to perform such a comparison will result in extra quality assurance in the diagnostic flow. It could also provide doctors with more confidence in that their diagnosis is correct.

The problem systems like this would face is to categorize medical reports in order to make further suggestions. One approach that is commonly used for such problems is machine learning, which is a field where a set of inputs is used to create a mapping to some output values [6]. This is done by using data to build a, usually statistical, model.

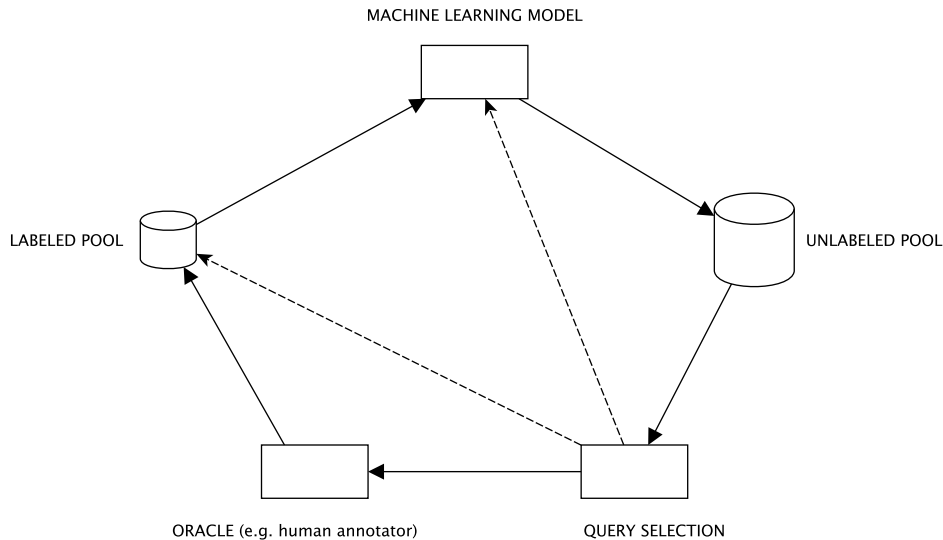


Figure 1.1: An overview of an active learning system. The dotted lines represents the things that some samplings strategies base their calculations on.

The task of predicting a category, or class, for a given text document is called text classification. Text classification is usually solved using supervised learning [1]. In supervised classification, there exists a set of inputs, in this case text data, that already have a category assigned to them. This data is then used to fit the model so that it later can make predictions for inputs that it has not yet been exposed to. Some models that have been shown to be successful in text classification are Neural Networks, Bayesian Classifiers, Decision Trees and Support Vector Machines (SVM) [1, 18, 1, 38].

In order to fit a machine learning model to predict categories for clinical reports, we need a set of already categorized data. That is, categories need to be assigned to an existing set of clinical reports. It is often the case that text data is widely available. Coming by data that is already categorized is on the other hand much harder. Obtaining high quality data is important for use in machine learning systems, both in healthcare and in other areas.

Since the models require a sufficient amount of categorized reports, the task of categorizing them can be cumbersome. Improving the categorization is especially important in the case of clinical data, since doctors and other clinicians time is both valuable and expensive. By improving the process and the quality of data to be labeled, and thereby reducing the number of reports that need to be categorized, they can spend more time doing their job.

Assigning categories to reports is often referred to as labeling. A field within machine learning that is focused on the task of improving the data labeling process is called active learning. It is a form of semi-supervised learning. The algorithm queries an oracle (in this case a doctor) for labels for the data points that it thinks will help the model improve the most. An overview of the components in an active learning system can be seen in Figure 1.1. This is used when there is plenty of readily available data, but assigning labels is expensive. Since the data points to be labeled are actively selected, the model can require fewer examples than if they were selected at random. The points can be selected by considering the certainty of the models, and request to label the documents that the model is less certain about. Another approach, which has not been given as much attention, is using the underlying structure of the data to select points. The goal with this approach is to capture the distribution of the categories by applying techniques such as clustering.

Figure 1.2: A screenshot of the interface used to label clinical reports at Sectra. A report is presented to the user, who can then select whichever categories (the blue buttons) deemed appropriate. By pressing “next”, the system presents a new reports.

If one of two categories is assigned to each document, it is called a binary classification problem [6]. Problems where one of several classes is assigned to an instance are called a multi-class classification problem. Multi-labeled classification is when one or more categories are assigned to each document. This thesis is mainly concerned with the multi-label case. Assigning several categories to a document is more time consuming than in the cases where only one option needs to be identified. In those cases the process can be stopped when the appropriate category has been found. However, when a document can be assigned several categories, the entirety of the text needs to be considered. For example, a news article can be on several subjects, such as both economics and sport. Even when the category sport has already been identified, the rest of the document still has to be read in order to find any additional categories. Using active learning to enhance the labeling of documents is therefore even more useful in the multi-labeled case, since the cost of labeling the individual reports is higher.

1.2 Aim

The purpose with this thesis project is to evaluate different active learning techniques that can be used to increase the quality of the labeled reports. Increasing the quality of the reports will also reduce the need for a large set of labeled reports within the project. Resulting from this will be a complete, standalone, system for labeling reports. The reports are interactively queried so a user can label the ones that are deemed most useful by the system. Labeling data to use in machine learning will probably be necessary for a long time ahead, but the aim here is to create a system that makes it more efficient. The technique chosen will then be used together with an existing web interface that Sectra created for the purpose of labeling reports. The existing web interface can be seen in Figure 1.2.

In the work that they have done so far in the research project a doctor has labeled an initial set of reports. This was done by simply selecting the reports in the order they were on file. The doctor that primarily worked with the labeling stated that the distribution over the

labeled categories was very skewed. The vast majority of labeled documents were assigned to a small subset of the categories. A skewed dataset causes the number of clinical reports that need to be labeled to increase a lot. For a statistical model to be able to achieve good results with the less frequently occurring categories, a large number of reports needs to be labeled in order to obtain a good amount of reports with these categories. Obtaining a more balanced dataset is desirable.

In addition to this, the doctors had come up with a new set of labels that were to be used. So the query strategies have to work from a non-existent, or very small, initial set of labeled reports.

1.3 Research questions

The specific research questions that this thesis treat are presented here. They are the main focus of study. There are two main research questions, both of which relate to active learning. In addition to this, a question regarding how to remove reports that does not need to be labeled is treated as well.

The clinical reports are of a multi-label nature. How we are choosing the documents to be sampled is important. Three different query strategies for active learning are evaluated for this purpose. These are: binary version space minimization, maximum loss reduction with maximum confidence and adaptive active learning. Their properties will be described in Chapter 2, and a motivation behind the choice of strategies will be made in Chapter 5. The algorithms to be evaluated will be based on the models certainty, as well as taking advantage of the underlying structure of the data through clustering. The first two questions treats the evaluation of the different strategies.

1. *How well does a machine learning model perform on the labeled data created based on the sampling strategies, given the constraints of the project?*

If the decision boundaries of the models can be used to pick documents that would be more informative for the model, the number of labeled documents could hopefully be reduced while still obtaining the same performance. When choosing the algorithm to use in the final system, there are several different constraints that will affect the final results, and therefore need to be taken into consideration. The number of initial reports needed for the strategies need to perform well, and the how many labeled samples need to be acquired to achieve good results will therefore be taken into account. In this regard, the strategies will be evaluated based on how well an SVM model perform on the data provided by them. This evaluation will be based on the accuracy, precision, recall and F_1 -score of the model. The labeling application is running on limited hardware, so the time it takes to run the strategies will also be evaluated.

2. *How do the the strategies affect the balance of labels in the labeled dataset?* Another indication on the quality of the labeled reports is the balance between the classes. Based on the initial sampling, the underlying distribution of labels in the clinical data is far from uniform. There are certain categories, like the one describing that everything is okay with the patient, that are a lot more common than other more rare illnesses.

If the dataset that is being sampled is skewed, i.e. some categories are a more frequent than others, our labeled set will likely follow that distribution. This will result in models requiring a lot of labeled documents to gather a sufficient amount of reports that are of the less common categories. Without these, the model will only perform well on the frequently occurring categories. Even though the original data may be imbalanced, selecting samples that contain a better balance between the different categories could improve the performance of the models. When a training set is imbalanced, the standard learning algorithms' performance can be significantly reduced [15]. The goal here is to

see which one of the different sampling techniques that will result in the best balance between the different categories in the resulting dataset.

In addition to this, after exploring the data it became clear that a substantial amount of reports states that an examination did not occur. There was no standard format to these, but their subject matter was very similar the third research question treats whether or not these reports can be filtered out using unsupervised machine learning techniques.

3. *To what degree is it possible to filter out invalid clinical reports by using an unsupervised techniques such as topic modeling?*

In the dataset from Sectra, there are reports describing patients not showing up for, or changing the time of their appointments, deceased patients or patients that have been ordered to go to another hospital. These reports do not contain any information of value from a medical point of view and should be discarded in the labeling process.


Unsupervised machine learning models such as topic modeling do by definition not require any labeled documents to train on. If it is possible to, without any such data, to capture the necessary variance and group these invalid reports together and they can be removed from the labeling process before a doctor is presented with them. That would result in an additional hurdle being removed from the process. The filtering of reports will be evaluated using accuracy, precision, recall and F_1 -score.

1.4 Delimitations

The active learning techniques will be evaluated on the publicly available Reuters dataset. Even though the sampling strategies are evaluated objectively on this, the applicability of the techniques on clinical data is only evaluated by one physician. That part of the study is therefore inherently subjective. Another limitation to this evaluation is that it is performed on the dataset provided by Sectra, which is not available for public use, so any conclusions drawn from it may not be applicable in other scenarios.

1.5 Structure of the Report

Chapter 2 covers the theory behind the different active learning strategies used, as well as some techniques used for classification and processing of text. After this, the data and an analysis of it is described in chapter 3, which is followed by Chapter ??, which presents the different experiments as well as their results. The method and results are then discussed in Chapter 5. Finally, Chapter 6 presents the conclusions and answers the research questions.



2 Theory

In this chapter the theory behind the techniques used in this thesis work will be presented. The first section will treat supervised text classification, which will act like a foundation for the active learning that will come after. Active learning is the main focus of the chapter, and an overview of the field as well as some concrete techniques will be covered. The techniques covered will mainly be in regards to multi-labeled data. In addition to this, some unsupervised methods used in the exploration of the data will be covered. This chapter will then end by going through the different evaluation metrics that will be use to assess the methods used.

Most of this Chapter deals with text data. Before using the techniques presented in this chapter, the text need to be represented in a way that is easy to work with. *Bag-of-words* (BoW) is one of the more common representations when performing text analysis. Using BoW the text is represented as a multi-set, a document is represented by the number of occurrences of the different words. Like the name implies, the positions of the words are not taken into account, they are viewed as if they were taken from a bag. One way to incorporate positional information into the representation is the use of n-grams. Instead of storing information pertaining to one term, information is stored with regards to n consecutive terms. Consider the text “Pattern Recognition and Machine Learning”. The use of an n-gram with $n=2$, called a bigram, would result in the tokens: “Pattern Recognition”, “Recognition and”, “and Machine”, and “Machine Learning”.

2.1 Text Classification

The problem in text classification is to assign one or more classes to a given text document [1]. It is mainly approached with supervised machine learning. That is, with a dataset that consists of a collection of text documents, where each document has one or more classes assigned to it. [6]. With the help of these labels, a classification model is fitted to the data. The goal of this is for the model to be able to correctly assign a class to a a previously unseen text document. Some of these classification models can also produce a probability of a document being of a certain class. Other models are based on the concept of a margin that separates the classes, and the distance between a data point and a margin can be used to indicate how certain the model is of the assigned label [38]. Example of use cases for text classification is categorization of news articles, document retrieval and email filtering. There exist several different models for classifying text. Decision trees, neural networks and Support Vector Machines (SVM)

are some have been previously applied to the text domain with successful results [2]. In this thesis, SVM are the main focus, since they have been studied extensively in the context of active learning [38, 34, 9, 42]. Logistic regression is used for the binary classification task of identifying invalid reports.

2.1.1 Support Vector Machines

SVMs work by implicitly map the training data to a feature space [6]. The goal is that the data should be linearly separable in the feature space, even if it is not in the input space. In the case of binary classification, a point is classified by the linear model:

$$y(x) = w^T \phi(x) + b \quad (2.1)$$

where the sign of $y(x)$ determines which label will be assigned to x .

The goal of an SVM is to try to find the hyperplane that maximizes the margin. That is, the distance between any point and the decision boundary should be as large as possible. A subset of the data points will be used to determine where the decision boundary is, these points are called the support vectors. The hyperplane that gives us the maximum margin can be found by:

$$\arg \min_w \frac{1}{2} \|w\|^2 \quad (2.2)$$

In order to allow for better generalization, and for data that is not completely linearly separable, SVMs make use of slack variables, denoted ξ_n . The slack variables are intended to penalize points that are close to the decision boundary [6]. A parameter $C > 0$ controls how much effect the slack variables will have. The equation with the slack variables becomes:

$$\arg \min_w \frac{1}{2} \|w\|^2 + C \sum_n \xi_n. \quad (2.3)$$

A smaller C -value allows more points to be misclassified, which is done in order to achieve better generalization.

Logistic Regression

Despite having regression in the name, logistic regression is a classification model. It is appropriate to use when there are categorical, binary, targets. Logistic regression works by determining the conditional probability of a class C_1 , given a feature vector ϕ . This probability can then be used with a certain threshold to determine whether or not a data point is a part of a certain class. Consider the case where we have two classes, C_1 and C_2 . Their probabilities are then calculated by [6]:

$$p(C_1|\phi) = \text{sigm}(w^T \phi) \quad (2.4)$$

where *sigm* is the *logistic sigmoid* function defined as:

$$\text{sigm}(a) = \frac{1}{1 + e^{-a}} \quad (2.5)$$

The probability for C_2 is then obtained with:

$$p(C_2|\phi) = 1 - p(C_1|\phi) \quad (2.6)$$

From Equation 2.4 it is clear that the number of parameters in the model is the same as the number of dimensions in the feature vector. Maximum likelihood is used in order to determine

the values of these parameters. For a dataset with the features x_n for $n = 1, \dots, N$ and targets $t_n \in \{0, 1\}$, the likelihood function can be written as:

$$p(\mathbf{t}|w) = \prod_{n=1}^N P(C_1|\phi)^{t_n} (1 - y_n)^{1-t_n} \quad (2.7)$$

where $\mathbf{t} = (t_1, \dots, t_N)^T$.

This is traditionally used with the Iterative Reweighted Least Squares method [6].

2.1.2 Multi-Label Classification

Multi-label classification is the type of text classification where one instance can be associated with multiple labels. It is a generalized version of the multi-class classification problem, where there is more than 2 labels, but each document is only assigned one [39].

A common way of solving multi-label classification problems is the Binary Relevance method [29, 8, 25]. It is a way of transforming the multi-label classification problem into several different binary ones. With Binary Relevance one classification model per label is fitted on the data. Each of these classifiers are then predicting whether or not the document is associated with the corresponding label or not.

2.2 Active Learning

Conventional machine learning systems use a set of available data to find a hypothesis that can explain the patterns within the data. The purpose of active learning is to allow a system to select the data that it wants to be labeled, and therefore the data it wants the model to be trained on [34]. An active learning system samples a document to be labeled from a pool of unlabeled data. With this sample an oracle, which is often a human annotator, is then queried to get the label for that document. By being able to decide what data to label and use, the goal is that the system can achieve better results, and that the training data will be of higher quality. To get a better understanding of how the process works, these are the steps that are commonly iterated over until enough samples have been labeled:

1. Evaluate the samples in the unlabeled pool based on a particular measure that is defined by the sampling strategy.
2. The selected samples are presented to an oracle that is queried for the labels. This oracle is commonly in the form of a human annotating the instances.
3. The newly labeled samples are added to the labeled pool.
4. A machine learning model is trained on the labeled pool, this model is often used by the sampling strategy in order to select samples to label in step 2

A model of the active learning system can be seen in Figure 1.1.

In several different domains, data is readily available and easy to obtain. But even if the data is abundant, labels for the data are often harder and more expensive to come by, especially when it comes to multi-label problems [34].

In Section ?? different ways to access the data in active learning are described This is followed by some theory of how the samples relate to the hypothesis space, in Section ??. The active learning theory then ends with a description of three multi-label active learning strategies.

2.2.1 Pool-Based Sampling

The main focus in active learning is how to select the samples to be labeled. There are different sampling methods in use, and which one is more appropriate depends on how the data can be accessed. Pool-based sampling is motivated by the assumption that there exist a large available pool of data, where only a small portion is labeled [22, 34]. The samples to be labeled are selected by evaluating the entire pool of unlabeled data, and choosing the most appropriate ones based on a defined utility measure. If the entire pool is too large, a subset could be used instead. For applied active learning, pool-based sampling seems to be the most popular choice [23], but there are some alternatives that have been used in theoretical settings such as stream-based selective sampling. The difference between stream-based selective sampling and pool-based sampling is that the draws one sample at a time from an input source and make the decision whether or not to query a label for it. For text applications, where a set of data is often readily available, pool-based sampling is often the more appropriate option since the entire dataset can be considered. Pool-based is therefore the sampling technique that is considered in this thesis.

2.2.2 The Version Space

In machine learning, a hypothesis is a specific configuration of a model, the purpose of which is to predict outputs on new instances of data by generalizing the training data. One hypothesis can, for example, be an SVM model with specific values for the parameters. The set of all possible hypothesis that we are working with is called the *hypothesis space*, denoted \mathcal{H} [30]. Following the SVM example, the hypothesis space would be the set of SVMs with the different parameter values under consideration. The subset of the hypothesis space that in the feature space separates the data is called the *version space*, which is defined as [34, 38]:

$$\mathcal{V} = \left\{ f \in \mathcal{H} \mid y_i f(x_i) > 0, \forall i \in \{1 \dots n\} \right\} \quad (2.8)$$

When the assigned label y_i , and the predicted label $f(x_i)$ has the same sign, $y_i f(x_i)$ will be positive, and therefore included in \mathcal{V} .

So the version space therefore represents the different hypotheses that make correct predictions on the training data. Under the assumption that one of the hypothesis can fully separate the data, the version space shrinks when more labeled data is acquired. So for new labeled instances the hypotheses in the version space will give better predictions for the training data. Based on this, an active learning algorithm should aim to reduce the size of the version space with each new sample. Optimally the selected sample should cut the version space in half in each iteration. By doing this it can be viewed as a sort of binary search, looking for the hypothesis that can fully separate the data.

There exists a useful relationship between the feature space \mathcal{F} and the hypothesis space \mathcal{H} called the *version space duality* [38, 41]. It states that hyperplanes in the hypothesis space corresponds to points in the feature space, and the other way around. So by selecting points to be labeled, constraints can be enforced on the hypothesis that form the version space.

One approach to this that has shown to be successful is *uncertainty sampling* [34]. This is commonly done with SVM, since the idea behind SVM is to find a hyperplane that separates two classes in a binary classification with the maximum margin. Out of the different hyperplanes in the hypothesis space, the version space contains those that can successfully separate the data. Uncertainty sampling aims to select the points in the feature space that will reduce the amount of the valid hypotheses the most. An SVM model tries to find the support vectors that maximizes the decision boundary in the feature space, separating the two classes. Considering this in \mathcal{H} , it will be analogous to the hypothesis in the center of the hypothesis space, which is encompassed by the constraints set by the labeled data. What uncertainty sampling predicts are the values for the unlabeled points, and then choosing the one that it is most uncertain about. The selected sample will therefore be the sample closest to the decision

boundary of the SVM. Based on the version space duality, it is a good approximation for dividing the version space in two [34].

2.2.3 Binary Version Space Minimization

Binary version space minimization (BSVM) is a generalization of uncertainty sampling, that is designed to make it work with multi-label data. The approach taken is to decompose the multi-label problem to several binary tasks with the binary relevance method, as is discussed in Section 2.1.2. The unlabeled point that is chosen for labeling is then the one with the smallest SVM margin across all the binary classification tasks. By doing this, it does not incorporate the multiple labels into the decision process, but treats all classes individually and equally. Selecting a new data x_{new} to be labeled using BSVM can formally be defined as:

$$x_{new} = \arg \min_{x \in D_U} \left(\min_{i \in \{1 \dots n\}} y_i(x) \right) \quad (2.9)$$

where n is the number of labels, $y_i(x)$ is the function from Equation 2.1 for the binary SVM classifier that is used to predict label i , and D_U is the set of unlabeled data points.

2.2.4 Maximum Loss Reduction with Maximum Confidence

Maximum loss reduction with maximum confidence (MMC) was developed by Yang et al [42]. The goal of this technique is to find the samples that will reduce the expected model loss the most, and select this sample for labeling. These are the basic notations that will be used when explaining the MMC approach:

- The labeled dataset: D_L .
- The unlabeled dataset: D_U .
- Possible query set: D_S .
- Optimal query set: D_S^* .
- The classification function that is trained on dataset D_L : f_{D_L} .
- A data point: x , and its label: y .
- The loss of on data point x : $L(f_{D_L}(x))$.
- The expected loss of the model: $\widehat{\sigma_{D_L}}$.

The expected model loss that MMC is trying to reduce can be defined as follows [42]:

$$\widehat{\sigma_{D_L}} = \int_x \left(\sum_{y \in Y} L(f_{D_L}(x)) P(y|x) \right) P(x) dx \quad (2.10)$$

It is hard to estimate $P(x)$, so it is instead measured over all the samples in D_U . This results in the estimate:

$$\widehat{\sigma_{D_L}} = \frac{1}{|D_U|} \sum_{x \in D_U} \sum_{y \in Y} L(f_{D_L}(x)) P(y|x) \quad (2.11)$$

After a set of data points D_S has been labeled, the new dataset $D'_L = D_L + D_S$ is obtained. Under the assumption that any $x \in D_U - D_S$ has an equal effect on a model trained on the datasets D_L and D'_L , we get the following equation for the reduction of the expected loss [42]:

$$D_S^* = \arg \max_{D_S} (\widehat{\sigma_{D_L}} - \widehat{\sigma_{D'_L}}) = \arg \max_{D_S} \left(\sum_{x \in D_S} \sum_{y \in Y} (L(f_{D_L}(x)) - L(f_{D'_L}(x))) P(y|x) \right) \quad (2.12)$$

In their paper, Yang et al. [42] considers the process of finding the greatest reduction in two steps: finding a good estimate for the conditional probability $p(y|x)$, and finding a way to assess the loss reduction of a multi-label classifier.

It is unfeasible for a query strategy to provide an estimation for all possible label combinations. If there are n different labels, there will be 2^n different label combinations. In order to estimate the conditional probability $p(y|x)$, MMC uses an approach that first estimates the number of labels for a given data point, and then uses that estimate to select the most probable labels. Consider the case where a data point has m labels. By obtaining the probability from our classification model for each label we can sort them in descending order. If the sample has m labels, then those should be the first m ones in the sorted list. Furthermore, the probabilities for these m labels are probably significantly higher than the probabilities for the labels that follow. There should be a clear separation between them.

Yang et al. [42] describe the process of estimating the number of labels as follows:

1. Use the classification model to obtain the probabilities for each label for all the data samples.
2. For each data sample, sort and normalize the probabilities for all the labels.
3. Using the labeled dataset, fit a logistic regression classifier with the sorted and normalized probabilities as features, and the number of labels as target.
4. With the fitted logistic regression model, predict the number of labels for the samples in the unlabeled pool.

After obtaining the predicted number of labels m for a sample x , the estimate for $p(y|x)$, denoted \hat{y} , is then obtained by selecting m the most probable labels based on the original classification models output.

now the loss for the multi-label classifier needs to be defined. By using the model where there are k different binary classifiers for a problem with k labels, the model loss can be calculated by adding the loss for the different binary classifiers like:

$$L(f) = \sum_{i=1}^k L(f^i) \quad (2.13)$$

where the loss of a single binary classifier is denoted as $L(f^i)$. With this definition, it remains to define the measure of loss on a single binary classifier. The measurement that is used by MMC is to estimate the model loss by the size of the version space of the SVM [38, 42]. The version space's size can be computed with Equation 2.8. However, computing this for each possible label combination is expensive. By using the heuristic from [37], an approximation of the version space with the added label can be obtained from the current SVM classifiers' margin. The reduction rate after adding the data point (x, y^i) , where $y^i \in -1, 1$, can be expressed as follows [42, 37]:

$$\frac{L(f_{D_L}^i + (x, y^i))}{L(f_{D_L}^i)} \approx \frac{V_{D_L + (x, y^i)}^i}{V_{D_L}^i} \approx \frac{1 + y^i f_{D_L}^i(x)}{2} \quad (2.14)$$

$L(f_{D_L}^i)$ does not involve the sample selected for labeling, so by writing the loss reduction as in Equation 2.15 it can be seen that focusing on the reduction rate is sufficient.

$$\begin{aligned} L(f_{D_L}) - L(f_{D_L'}) &= \sum_{i=1}^k (L(f_{D_L}^i) - L(f_{D_L'}^i)) \\ &= \sum_{i=1}^k (L(f_{D_L}^i) (1 - \frac{L(f_{D_L'}^i)}{L(f_{D_L}^i)})) \end{aligned} \quad (2.15)$$

By incorporating the result from Equation 2.14, the following approximation for the reduction rate is obtained:

$$\sum_{i=1}^k \left(\frac{1 - y^i f_{D_L}^i(x)}{2} \right) \quad (2.16)$$

The only thing that remains is now to combine the estimation of $p(x|y)$ with the estimate for loss reduction. The resulting equation, called maximum loss reduction with maximal confidence, is [42]:

$$D_S^* = \arg \max_{D_S} \left(\sum_{x \in D_S} \sum_{i=1}^k \left(\frac{1 - \hat{y}^i f_{D_L}^i(x)}{2} \right) \right) \quad (2.17)$$

The full algorithm for the approach can be seen in Algorithm 1.

Algorithm 1 Maximum Loss Reduction with Maximal Confidence procedure. Taken from Yang et al. [42], with some modifications to the notations used in order to make it coherent with the rest of the report.

Input: Labeled set D_L

Unlabeled set D_U

Number of classes k

Number of selected examples per iteration S

repeat

Train k binary SVM classifiers f^1, \dots, f^k based on training data D_L .

for each $x \in D_U$ **do**

Predict its label vector using the method described in Section 2.2.4.

Calculate the expected loss reduction with the most confident label vector \hat{y} , $score(x) =$

$$\sum_{i=1}^k \left(\frac{1 - \hat{y}^i f_{D_L}^i(x)}{2} \right)$$

end for

Sort $score(x)$ in decreasing order for all x in D_U .

Select a set of S examples D_S^* with the largest scores, and update the training set $D_L \leftarrow D_L + D_S^*$.

until enough instances are queried

2.2.5 Adaptive Active Learning

In their paper, Li et al. [23] present two approaches to active learning:

- Max-Margin Uncertainty Sampling
- Label Cardinality Inconsistency

These two techniques are then combined in a weighted fashion into what they call Adaptive Active Learning (AAL).

Max-Margin Uncertainty Sampling

The idea behind *Max-Margin Uncertainty Sampling* comes from the observation that multi-label classification prediction is mainly about separating the positive labels from the negative labels [23]. That is, separating the labels are assigned to an instance and the ones that are not. In order to model the uncertainty of the prediction on a datapoint, Li et al. suggest the usage of a global separation margin to separate the negative labels from the positive.

The positive labels for a data point x is defined as those where $sign(f'_{D_L}(x))$ is positive. The separation margin is defined as:

$$sep_margin(x) = \min_{i \in \hat{y}^+} |f'_{D_L}(x)| + \min_{i \in \hat{y}^-} |f'_{D_L}(x)| \quad (2.18)$$

where \hat{y}^+ denotes the set of predicted labels that are positive on the instance, and the \hat{y}^- denotes the negative ones.

The data point that the model is the most uncertain about is then the one with the smallest margin. Li et al. define their global measure, max-margin prediction uncertainty, as:

$$u(X) = \frac{1}{\text{sep_margin}(X)} \quad (2.19)$$

Label Cardinality Inconsistency

Label Cardinality Inconsistency is based on the assumption that the underlying distribution is the same for the labeled and unlabeled data. In a multi-label dataset, the *label cardinality* is defined as the average number of labels assigned to each class [39]. The selection strategy that Li et al. based on this measures the Euclidean distance between the number of assigned predicted labels on x , and the label cardinality of the labeled data:

$$c(x) = \left\| \sum_{i \in \hat{y}^+} 1 - \frac{1}{N_L} \sum_{y \in Y_L} \sum_{i \in y^+} 1 \right\|_2 \quad (2.20)$$

where N_L is the number of labeled samples, Y_L are the labels for those samples, and y^+ are the positive labels in y .

Integration - Adaptive Active Learning

Since *max-margin uncertainty sampling* and *label cardinality inconsistency* complement each other, an integration method is used:

$$q(x, \beta) = u(x)^\beta \cdot c(x)^{1-\beta} \quad (2.21)$$

where β is a parameter controlling the weight put on the two measures. This parameter is chosen by in each iteration evaluating a discrete set of values, for example $\{0, 0.1, 0.2, \dots, 1\}$. The selection of β is then based on the most informative sample amongst them. Equation 2.21 shows the *approximate generalization error*, which is used to select the sample.

$$\epsilon(x) = \sum_{x \in D_U} \max_{i \in f(x)^+} (1 - f^i(x)) + \max_{i \in f(x)^-} (1 + f^i(x)) \quad (2.22)$$

,where $f(x)^+$ and $f(x)^-$ are the predicted positive labels, and negative labels, respectively. So, the sample is then selected by:

$$x^* = \arg \min_{x \in D_U} \epsilon(x) \quad (2.23)$$

The complete algorithm for the integrated approach can be seen in Algorithm 2.

2.3 Text Processing using Unsupervised Techniques

Techniques in machine learning that do not require a set of categorized data is called unsupervised learning. They use the structure of the data to obtain the information to use when processing it. These techniques may have different goals. Some are used to estimate a distribution, others are used to reduce the number of dimensions in the data by trying to find dimensions that capture as much as possible of the variance, and some are used for the purpose of identifying groups of similar points within the data [6]. When it comes to text data, there are a few common methods and techniques that are unsupervised, and can be used for different purposes. Examples of such techniques are *topic modeling* and *clustering*. Another interesting technique is word2vec, that is used to produce word embeddings.

Algorithm 2 AAL Procedure. Taken from Li et al. [23], , with some modifications to the notations used in order to make it coherent with the rest of the report.

Input: Labeled set D_L
 Unlabeled set D_U
 Parameter set B

repeat
 Train multi-label SVM classifier F^0 on D_L .
 for each $x_i \in D_U$ **do** **do**
 Compute $u(x_i)$ and $c(x_i)$.
 end for
 for each $\beta \in B$ **do** **do**
 Mark a candidate instance $x = \arg \max_{x \in D_U} q(x, \beta)$
 end for
 Copy all marked candidate instances into a set S .
 for each $x \in S$ **do** **do**
 Produce \hat{y} using classifier F^0 .
 Retrain a new classifier F on $(x, \hat{y}) \cup D_L$.
 Compute $e(x)$ using classifier F and Equation 2.22.
 end for
 Select instance x^* from S using Equation 2.23
 Remove x^* from D_U , query its label vector y^* .
 Add (x^*, y^*) to D_L .
until enough instances are queried

As described in Section 2.1, with BoW a document is represented by the number of occurrences of the different words. Representing text in this way therefore becomes high-dimensional, there is one dimension for each word in the vocabulary. In written language a word can be used to express several different thoughts, and one thought can be expressed using several different words. This is something that cannot be captured with BoW. However, it is easy to work with, and is commonly used when performing topic modeling among other things.

2.3.1 Topic Modeling

A topic model is a statistical model for finding topics within text [11]. The topics build upon the probability that a certain word would occur in a text about a given topic, on the basis of terms occurring together. For example, if the topic represents United States politics, words such as “government”, “Trump”, “Reagan”, “Senate”, or “Medicaid” are more likely to appear than “sailboat” or “sweater”. Any given document can then contain a certain topic with some probability. This can be viewed as fuzzy clustering, and that the document has a degree of membership in a topic or cluster [11]. The most common topic model in use today is Latent Dirichlet Allocation (LDA) [11]. Another topic model that preceded LDA is Probabilistic latent semantic analysis (PLSA) [16]. However, PLSA has been shown to be more prone to overfitting than LDA [11].

In the rest of the report, the following notation will be used:

- D denotes a corpus of M documents: $D = \{w_1, w_2, \dots, w_M\}$.
- The number of topics is K . Each topic is indexed by i .
- N_d is the number of terms in document d .
- N_i is the number of terms n in topic i .
- V denotes the number of words in the vocabulary.

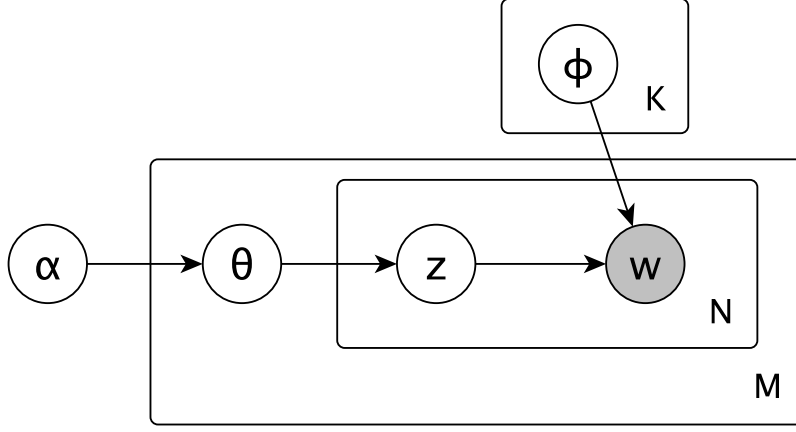


Figure 2.1: Diagram of the LDA model.

Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) is a statistical model, where abstract topics in the model are defined as distributions over words [7]. LDA is based on a generative process, a model of which can be seen in Figure 2.1. The circles in this figure represent random variables. Dependencies between these random variables are shown with arrows, and if a variable is observed it is shaded in the figure. In this model, the only observed variable is the words in the document. Parts of the model are surrounded by a rectangle to show that the part is repeated several times.

The generation of a corpus is done with the following steps [11, 7]:

- **Draw a distribution over the words for each topic.** A sample Φ_i is drawn from a symmetric Dirichlet distribution with parameter β . This sample represents the distribution of terms for the topic i .

$$\Phi_i \sim \text{Dir}(\beta) \quad (2.24)$$

$$p(\Phi_i|\beta) = \frac{\Gamma(V\beta)}{\Gamma(\beta)^V} \prod_{v=1}^V \phi_{iv}^{\beta-1} \quad (2.25)$$

Here, Γ is the gamma function, and ϕ_{iv} is the value for a word v in the topic i .

- **Draw a distribution over the topics for each document.** A sample θ_d is drawn from a Dirichlet distribution with parameters α . This sample represents the distribution of topics for document d .

$$\Theta_d \sim \text{Dir}(\alpha) \quad (2.26)$$

$$p(\Theta_d|\alpha) = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^K \Gamma(\alpha_i)} \prod_{i=1}^K \theta_{di}^{\alpha_i-1} \quad (2.27)$$

Here, θ_{di} is the probability of topic i for document d .

- For each token with index n :

- **Draw a topic assignment z_{dn} for the token index n .** z_{dn} is drawn from the distribution over topics for each document. That is, z_{dn} is drawn from a multinomial distribution using θ_d as a parameter.

$$z_{dn} \sim \text{Multinomial}(\Theta_d) \quad (2.28)$$

$$p(z_{dn} = i | \Theta_d) = \theta_{di} \quad (2.29)$$

- **Draw a token w_{dn} .** The token w_{dn} is drawn from the topic distribution assigned to the index n . That is, w_{dn} is drawn from a multinomial with parameter $\phi_{z_{dn}}$.

$$w_{dn} \sim \text{Multinomial}(\Phi_{z_{dn}}) \quad (2.30)$$

$$p(w_{dn} = v | z_{dn} = i, \Phi_i) = \phi_{iv} \quad (2.31)$$

The LDA model identifies topics from different terms that occur in the same document. Consider the case where an LDA model has been used to learn a number of topics. Two terms that frequently occur together are then likely to be in the same topic. So, if the same word has been used to express different thoughts, and the word has the same probability in two topics, the words that it co-occurs with can be used to differentiate between the different thoughts.

The task of learning the LDA model is a Bayesian Inference problem. We have several variables that we cannot observe: the word distribution for the topics (ϕ_i), the topic assignments for the tokens (z), and the topic distribution for the documents (θ_d). The only observed variables are the words in the document. We have to approximate the posterior distribution using some sampling method, since it cannot be inferred automatically [7].

There exist a few algorithms that can be used to learn topics for the LDA model. Two of these that has shown to be able to extract useful topics from text are *collapsed Gibbs sampling* [14] and *variational Bayes* [7]. Variational Bayes works by using simple single-variable models to approximate the LDA. As a consequence, it disregards any dependencies between the variables.

Collapsed Gibbs Sampling

Gibbs sampling is a Markov chain Monte Carlo (MCMC) method that is often used to obtain a good estimate for the distribution of a probability model, when it is not feasible to sample the distribution directly [11]. With the help of a heuristic, or randomly, Gibbs sampling initializes the variables. During a large number of iterations the variables are then sampled. When a variable is being sampled, it is conditioned on the others. In an MCMC fashion, a number of samples are rejected during an initial burn-in period. This is done in order to get to a state where the points are more representative of the distribution that is estimated.

Griffith et al. [14] came up with collapsed Gibbs sampling for the LDA model, where θ and ϕ are marginalized out. The only variable to then be repeatedly sampled is the topic assignment, z_{dn} , conditioned on the assignments of the other tokens.

2.3.2 Text Clustering

Cluster analysis is commonly defined as finding groups in a given dataset. The members of these groups are determined to be similar by a similarity measure [19, 2]. Text data are sparse, but yet have a very high dimensionality. With one dimension per term in the dictionary, it is not uncommon with dimensions in the order of 10^5 . For this reason, some of the more naive clustering algorithms do not work well for text data [2].

In distance-based clustering, a similarity function is used to measure the closeness between two text documents. For the purpose of measuring the similarity between text objects, the

cosine similarity function is commonly used, as well as Euclidean distance [2]. Given two n dimensional points a and b , the definitions for the two can be seen in Equation 2.32 and Equation 2.33. Two different approaches to distance-based clustering are distance-based partitioning, and agglomerative hierarchical clustering. For the distance-based approach, k -means and k -medoid are two frequently used algorithms.

$$\text{similarity}(a, b) = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2} \sqrt{\sum_{i=1}^n b_i^2}} \quad (2.32)$$

$$d(a, b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2} \quad (2.33)$$

k -means Clustering

When using the k -means clustering algorithm, the clusters are based upon an initial set of k representatives. A simple approach to k -means clustering could look as follows:

1. Select K seeds from the original dataset
2. Assign the rest of the documents to one of these seeds, based how how similar they are by the similarity function
3. Before each new iteration, select a new centroid for each cluster. This should be the point that is the best central point for the cluster.
4. Repeat step 2 and 3 until convergence.

A visualization of this can be seen in Figure 2.2. One advantage that k -means has over K -medoid is that it requires a small number of iterations, especially compared to K -medoid [2, 33]. However, k -means is rather sensitive to the selection of initial seeds. One approach is to just select them randomly, or selecting them based on the result of another lightweight clustering method. A frequently used method is k -means++, that has been shown to improve both the speed and accuracy of k -means clustering [4].

2.3.3 Word Embeddings

Word embeddings can be done with several different techniques, and it is the process of representing a word as a real-valued vector instead of just an atomic unit. Viewing a word as a vector allows for doing interesting things with them, such as evaluating how similar to words are. Evaluating the similarity of words that is hard to do when treating them as atomic units.

This can be created by using a co-occurrence matrix to see how often certain words occur together, and then perform some dimensionality reduction on them [20, 21]. Another approach that shown to be very successful in producing high quality word embeddings is *word2vec*, which uses a neural network to accomplish this task [26]. In addition to being able to compute similarities between words, using simple algebraic some interesting relationships can be discovered. The example that Mikolav et al. [26] showed was that using the vector for “King”, subtracting the vector for “Man” and adding the vector for “Woman” resulted in a vector that was close to that representing “Queen”.

One approach to this is using a continuous bag-of-words model [26]. A neural network is used to predict the middle word using both words occurring before and after it. The four words occurring before and after the middle word is used as inputs, but their internal order is not used. This is the reason for the name, continuous bag-of-words.

The second approach that Mikolav et al. explored was a continuous skip-gram model [26]. Here a neural network with a continuous projection layer was used. With the current word as input, co-occurring words within a certain range are predicted. A bigger range is more computationally complex, but results in word vectors of higher quality.

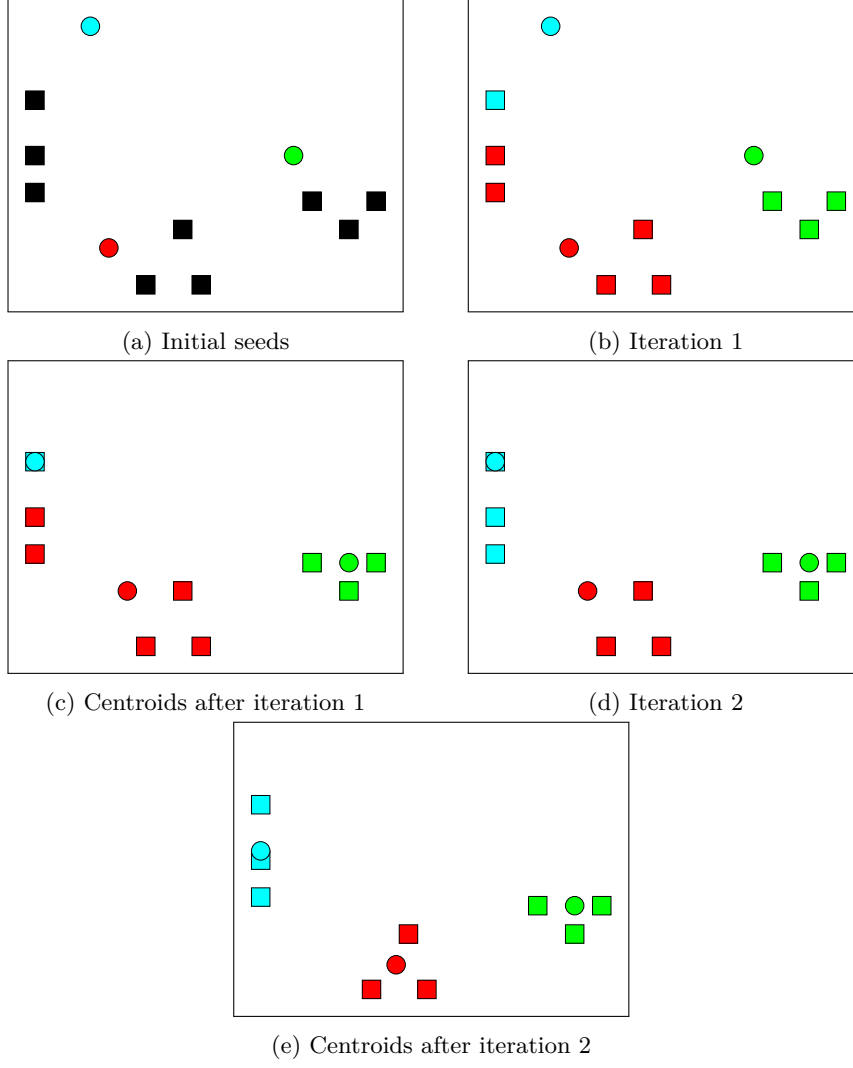


Figure 2.2: (a) to (e) shows iterations of k -means until convergence. In (e) it can be seen that the new centroids capture the same documents as the previous iteration, and we have converged. The circles represents the seeds for the clusters and the data points are represented by squares. The color of the points are shows which cluster the point is currently assigned to.

Word2vec does not require labels to be provided with the data, but uses the data itself to generate targets. For this reason it is sometimes called a self-supervised technique.

2.4 Evaluation Metrics

For classification or information retrieval systems, the typical evaluation metrics in use are *precision*, *recall* and *recall* [17]. We define the following metrics in terms of *true positives*, *false positives*, *true negatives* and *false negatives*. How they are defined can be seen in Table 2.1. Data points that are correctly classified are then either *true positives* or *true negatives*.

Precision is the percentage of the results found by the system that are correct [31]. Recall is the percentage of correct results in the dataset that are found by the system. Precision and recall are defined as follows:

$$Precision = \frac{tp}{tp + fp} \quad (2.34)$$

	Correct P	Correct N
Predicted P	True Positive	False Positive
Predicted N	False Negative	True Negative

Table 2.1: Confusion matrix for explaining true positives, false positives, true negatives and false negatives

$$Recall = \frac{tp}{tp + fn} \quad (2.35)$$

F-score is the harmonic mean between recall and precision, and is defined as [31]:

$$F = \frac{2 * precision * recall}{precision + recall} \quad (2.36)$$

Another metric that is used for evaluation is *accuracy*, which is the percentage of predictions that matches the actual labels. Accuracy is defined as follows:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \quad (2.37)$$

These metrics are designed to work when there are two labels. Given several labels, there are different average methods used to get a score for the system. Two of these are *micro* and *macro* averaging.

The number of true positives, false positives, true negatives and false negatives for an instance λ are here denoted as tp_λ , fp_λ , tn_λ and fn_λ . A binary evaluation measure on these is denoted as $B(tp_\lambda, fp_\lambda, tn_\lambda, fn_\lambda)$. Micro-average work by summing the individual true positives, false positives, true negatives and false negatives [40]. Then you use the sums to obtain the final score. Using the notation established above, the micro average is defined as [40]:

$$B_{micro} = B\left(\sum_{\lambda=1}^k tp_\lambda, \sum_{\lambda=1}^k fp_\lambda, \sum_{\lambda=1}^k tn_\lambda, \sum_{\lambda=1}^k fn_\lambda\right) \quad (2.38)$$

Macro-average on the other hand works by first calculating the binary measure, and then taking the average of all of them [40]. It is defined as:

$$B_{macro} = \frac{1}{k} \sum_{\lambda=1}^k B(tp_\lambda, fp_\lambda, tn_\lambda, fn_\lambda) \quad (2.39)$$

It is worth noting that for some measures, such as Accuracy, the result of the two averaging approaches is the same. However, it differs for *recall* and *precision*, and therefor also the *F1-score* [40].

Perplexity can be used in order to compare different probabilistic models. It is a measurement that determines how good a models predictions are, where a lower score means that the model is better at predicting. By evaluating the perplexity on a test set, it will give an indication of how well the model will generalize [7]. Perplexity for a set of M documents, on a dataset D is:

$$perplexity(D) = \exp \left\{ - \frac{\sum_{i=1}^M \log p(x_i)}{M} \right\} \quad (2.40)$$



3 Data

This chapter contains a description of the two datasets used, together with the preprocessing steps and a description of how the data was represented. It also provides a more in depth analysis of the relationship between certain aspects of clinical dataset and the topics generated by an LDA model. These relationships perform the basis for some of the decisions made in Chapter 4.

3.1 Datasets

Two different datasets were used in this thesis. They were the dataset of clinical reports provided by Sectra, as well as Reuters-21578 ¹. The latter was used in order to be able to simulate a multi-label labeling process. Before being integrated into Sectra's system, the different active learning strategies needed to be evaluated from an objective point of view, so that any tradeoffs were known beforehand. Since the vast majority of the dataset from Sectra was unlabeled, this could not effectively been done using only that. In order to simulate an annotator labeling reports, the unlabeled pool must consist of data points for which a label can be automatically retrieved. Since the Sectra dataset had only had 500 assigned labels, this would not have provided as much insight as the Reuters-21578 dataset.

3.1.1 Clinical Dataset From Sectra

The set of reports provided by Sectra contained 1068904 different entries, where 493 were initially labeled. The entries were spread out over several files and stored in the JSON format. However, those labels were subject to change, so they were mainly used to see if there was a correlation between the labels and clusters during the exploration phase. A sample report can be seen in Figure 3.1. The fields include:

1. **ExamId**: The ID of the exam.
2. **ReportText**: The text for the report written by the physician after the examination.
3. **Anamnesis**: The patient's medical history.

¹Reuters-21578, <https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection>

```

{
  'ExamId':      3302250,
  'ReportText':  '[NUM-SEQ]
                  Craniell datortomografi utan och med
                  intravenös kontrast:

                  Frontalt på höger sida finns ett c:a
                  4 x 3 cm stort lågattenuerande område, tolkas
                  representera rest efter genomgången
                  parenchymskada, troligen äldre kontusionsblödning.
                  Subcorticalt på ömse sidor om centralfåran på
                  höger sida finns ett några cm-stort
                  lågattenuerande område som kan vara ischemiskt.
                  Lätt sänkt attenuering av vit substans
                  periventrikulärt förenlig med leuko-araios av
                  degenerativ natur. För åldern normalstora
                  ventriklar. Corticala sulci upp mot konvexiteten
                  är något smalare än förväntat för åldern.

                  Någon tumörsuspekt förändring påvisas ej.',
  'Canceled':    False,
  'Question':    'Förändring vä temporalt?',
  'PatientAlert': 'Hepatit C-positiv.',
  'ExamComment': 'Alla kontrastfrågor: UA mnn',
  'ExamName':    'DT hjärna utan och med iv kontrast',
  'Anamnesis':   'Pat med skalltrauma på 60-talet. Kommer nu med
                  nattliga från-varoattacker. Skrikigt beteende
                  som tolkats som epilepsi. CT är aldrig gjort.
                  HEPATIT C-positiv."',
  'ExamCode':    '81081',
  'PatientSex':  'MALE',
  'PatientAge':  59,
  'Urgent':      0,
  'Pharma':      [{"ExamId": 3200240, "Units": "100 ml",
                  "Pharma": "Omnipaque Inj.lösn 300 mg I/ml"}]
}

```

Figure 3.1: A sample report from the dataset provided by Sectra

4. **PatientAlert:** Anything special about the patient.
5. **ExamComment:** Comments regarding the performed examination.
6. **Cancelled:** Whether or not the examination was Cancelled.
7. **ExamName:** Name of the exam.
8. **ExamCode:** Code for the exam.
9. **PatientSex:** The sex of the patient.
10. **PatientAge:** Age of the patient. This field is truncated if it is above 90 years.
11. **Urgent:** If the examination is urgent or not.
12. **Pharma:** List of administrated pharmaceuticals.

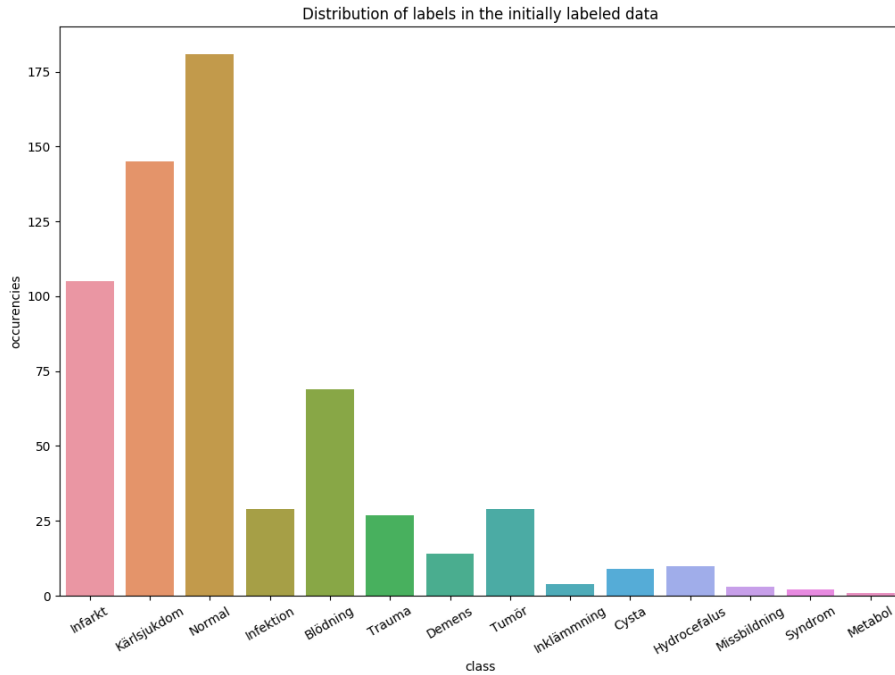


Figure 3.2: The distribution over the labels in the initial set of labeled data provided by Sectra

The work was mainly concerned with the ReportText field, since it contains the response to the result of the examination. But for the complete active learning system the Anamnesis field was used as well. The labels that were initially assigned to these reports were: “Blödning”, “Infektion”, “Metabol”, “Tumör”, “Cysta”, “Missbildning”, “Syndrom”, “Demens”, “Hydrocefalus”, “Infarkt”, “Kärtsjukdom”, “Trauma”, “Systemsjukdom”, “Inklämmning” and “Normal”. The distribution of categories among these initially labeled reports can be seen in Figure 3.2. Note that this is only a count of the different labels. The plot is therefore disregarding which labels occurred together.

3.1.2 Reuters-21578

The Reuters-21578 newswire dataset is widely used when it comes to text classification research, and provides a good multi-label benchmark that can be used to compare how well certain techniques perform to other papers. It is a set of news stories, so there is only one text data field. All experiments used the *ModApte* split of the dataset, which is commonly used and readily available. It splits the dataset into a predefined set of training and test documents, containing 7769 and 3019 entries respectively. This split contains a subset of the categories, specifically 90 different ones. Since the clinical dataset from Sectra only contained 15 different categories, this would not mirror that very well, so instead the 15 most common categories of those were taken out. The 15 most common categories were (in order of label count): “earn”, “acq”, “money-fx”, “grain”, “crude”, “trade”, “interest”, “wheat”, “ship”, “corn”, “money-supply”, “dlr”, “sugar”, “oilseed” and “coffee”. The distribution of the top 15 Reuters-21578 categories can be seen in Figure 3.3. After filtering out the documents not labeled with any of the top 15 categories, there were 6880 documents left in the training set, and 2646 in the test set.

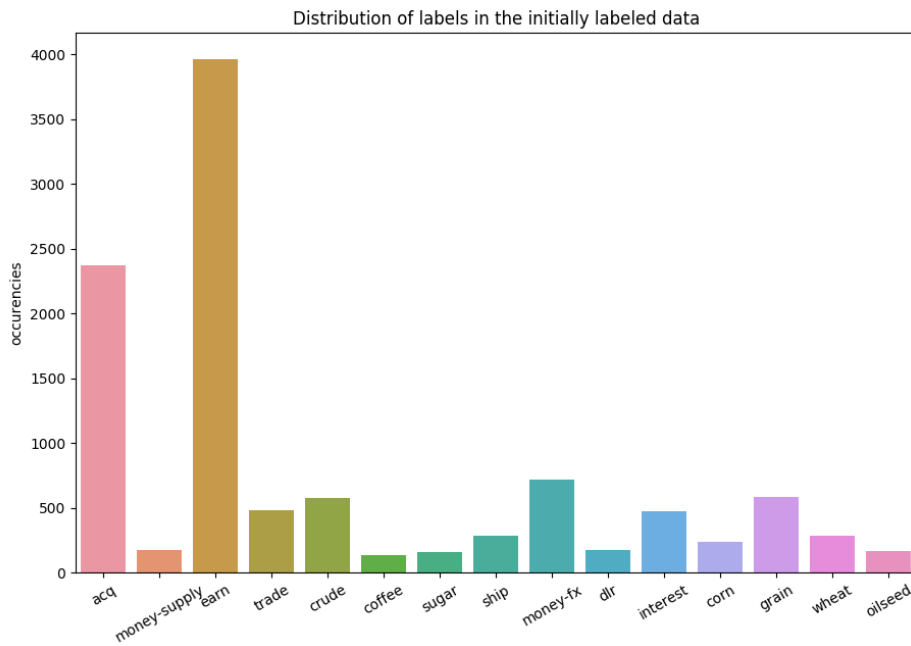


Figure 3.3: The distribution over the labels in the Reuters data

3.2 Pre-Processing and Text Representation

Before the data was used in the conducted experiments, several pre-processing steps were applied in order to clean the dataset and make it easier to work with. They were:

1. The first step was to extract the fields of interest. For the exploratory phase, and for the use of active learning techniques these were “ReportText” and “Anamnesis”. When it came to filter out invalid reports, the “ReportText” was the only field of concern. It describes the results of the examination, and therefore if there was an examination at all.
2. White space and punctuation were stripped from the data.
3. All words were transformed into lowercase.
4. The most common words as well as very infrequent words were both filtered out. Specifically, words occurring in less than 1% of the documents, and words occurring in more than 90% were removed. The idea behind this is that these words would not contribute to differentiating different classes of documents. Removing of both frequent and infrequent words is commonly done when working with text and has been done in the context of classification, active learning or topic modeling before [38, 7, 9, 32].
5. A list of identified common stopwords were removed as well. This list of words was based on the Swedish nltk stopwords list. After iterating over the dataset words that occurred frequently but were not considered to be very informative for the models were identified. The list of stopwords was then extended to incorporate these words as dataset-specific information. For example, this included names of the doctors that had written the report. By removing names of doctors the idea is to make the system more applicable to new reports, written by other doctors.
6. Accents from the words were removed.

Original	Replacement	Type
ordinärt	normalt	synonyms
ej	inte	synonyms
avbeställd	avbokad	synonyms
avebställd	avbokad	misspelling + synonym
belsutat	beslutat	misspelling
måttliga	lätta	synonyms
pat	patient	short
pt	patient	short
pateint	patient	misspelling
akuten	akutmottagningen	misspelling
us	undersökning	short

Table 3.1: The synonyms, misspellings and shorts found in the data that the author could assert with confidence.

7. The text was tokenized and then stemmed using the Swedish Porter2 stemmer ².

Most of these steps have been performed in previous research dealing with text analysis in the form of classification or active learning [38, 7, 9, 32].

Identifying synonyms and names that could be added to the stopwords was done with a word2vec model. A word2vec model was used on the entire dataset to analyze the relationship between terms and to find possible synonyms. In order to find synonyms, all words in the dataset that according to the word2vec model had a similarity over 95% were manually inspected. By doing this, 420 pairs were discovered. The vast majority of these were words that are used in similar contexts, which includes opposites like “left” and “right”, and some names. Some of the medical terms were hard to interpret, and were therefore not considered to be synonyms. Disregarding these, the synonyms and misspellings that were decided to be used in the final system can be seen in Table 3.1. The original value was replaced with the new one during the preprocessing stage.

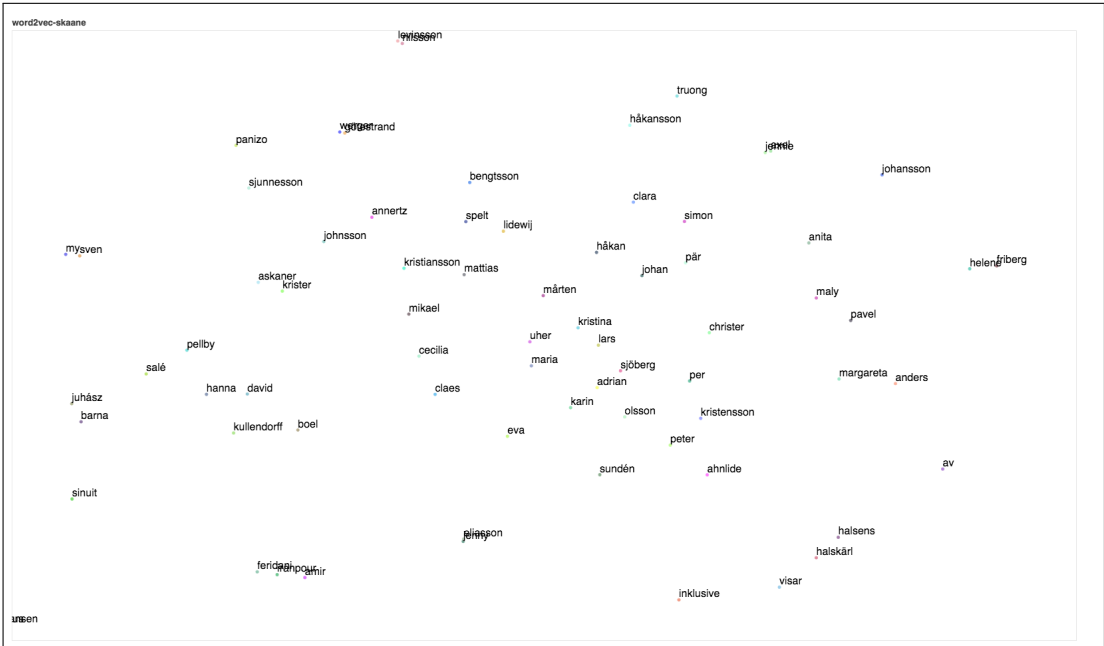
In addition to finding synonyms, this model was used to identify names and other identifiers in the reports. They would come up as similar entities by the model, since they are often used in the same context. In order to visualize the data in a 2D plot, t-distributed stochastic neighbor embedding (t-SNE) was used. It is a dimensionality reduction technique, that is able to transform high dimensional data into two dimensions, while working to retain as much variance as possible. For the purpose of identifying names, a key insight was that most of the names in the medical reports were used in very similar contexts. Usually it was a doctor providing a signature to the examination. Since names are commonly used in similar contexts, they have similar attributes in the word embedding model. Given that the names got similar coordinates in the plot, identifying the section with names allowed for identification of a lot of the names used in the reports. Figure 3.4 shows how this was used to identify a group of names.

After transforming the text into a sequence of tokens, the final step before using it with the models was to create a representation that would be beneficial to work with. The representation chosen was bag of words, i.e. a matrix of tokens count. Each document is represented by the counts of each token, disregarding the order of the tokens. In order to get some positional information into the representation additional tokens are stored. The additional tokens are bi-grams, which are pairs of tokens (i.e. processed words). By storing the frequency of how often such a pair occurs in the document, alongside the regular one word tokens, some positional information is retained.

²Swedish Porter Stemmer, <http://snowball.tartarus.org/algorithms/swedish/stemmer.html>



(a) A 2D plot of the full word2vec model.



(b) A 2D plot of the subset of the word2vec model covering the names.

Figure 3.4: Two figures illustrating how the names were discovered in the word2vec plot. The (b) plot represents the red square in (a).

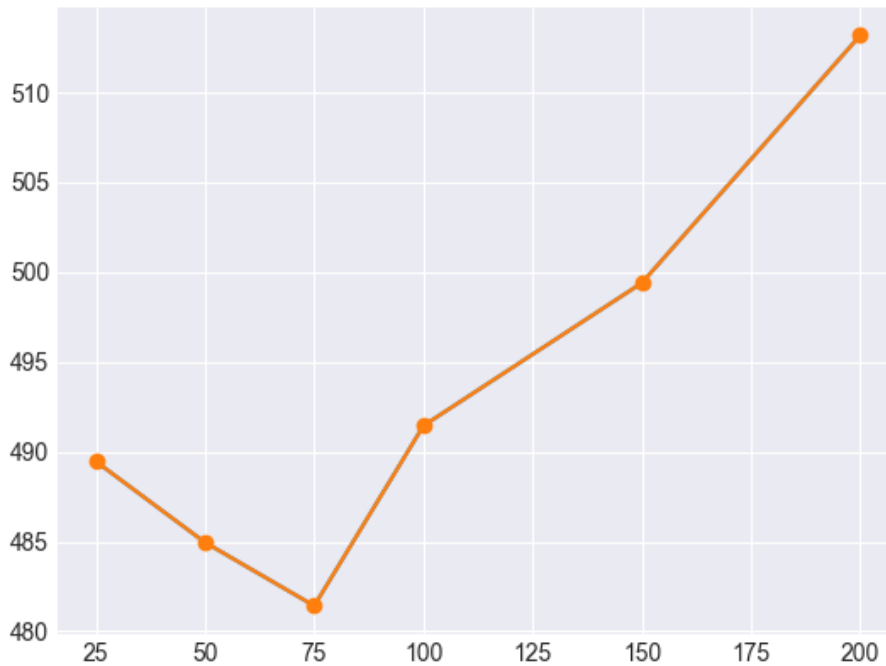


Figure 3.5: The perplexity scores for the different LDA models

3.3 Exploratory Study

For the exploratory study we used the representation described in Section 3.2. The goal with this phase was to acquire a better understanding of the data, how it was structured and what kind of information might be extracted from it. A part of this goal was to go through the fields for the different reports to see how they worked and what values could be expected. Certain fields such as the canceled field did not seem to be very reliable. Reports that clearly explained a situation where the patient had been transferred to another hospital, or for another reason not having performed an examination, still described a situation where the canceled field was set to “false”. This is treated in research question 3.

The first step was to fit an LDA model to the data. Different topic models were tested for this purpose. The topic model used was Latent Dirichlet Allocation. In order to use the LDA model the number of topics, k , has to be selected, as described in Section 2.3.1. The different values of k that were evaluated was 25, 50, 75, 100, 150, 200. To determine which topic model that should be used in the exploration, their perplexity was compared and the models with the lowest perplexity was chosen. Perplexity is used in the original LDA paper by Blei et al. [7] to compare different number of topics. Hofmann used it to evaluate the pLSI based topic models as well [16]. 80 000 reports were used in the experiment, and they were selected at random. The reason for only using a subset is that the number of reports available would be too big to use in the final active learning system due to performance constraints. The models were fitted on 72 000, or 90%, of these reports, and the additional 10% were used as a held-out set to evaluate the models. Perplexity for the evaluated models can be seen in Figure 3.5. Based on this, the selected model was the LDA model with 75 topics.

The data points were plotted in a 2D plot after reducing the dimensions using t-SNE. Each data point was colored based on some trait that the specific data point had. For the

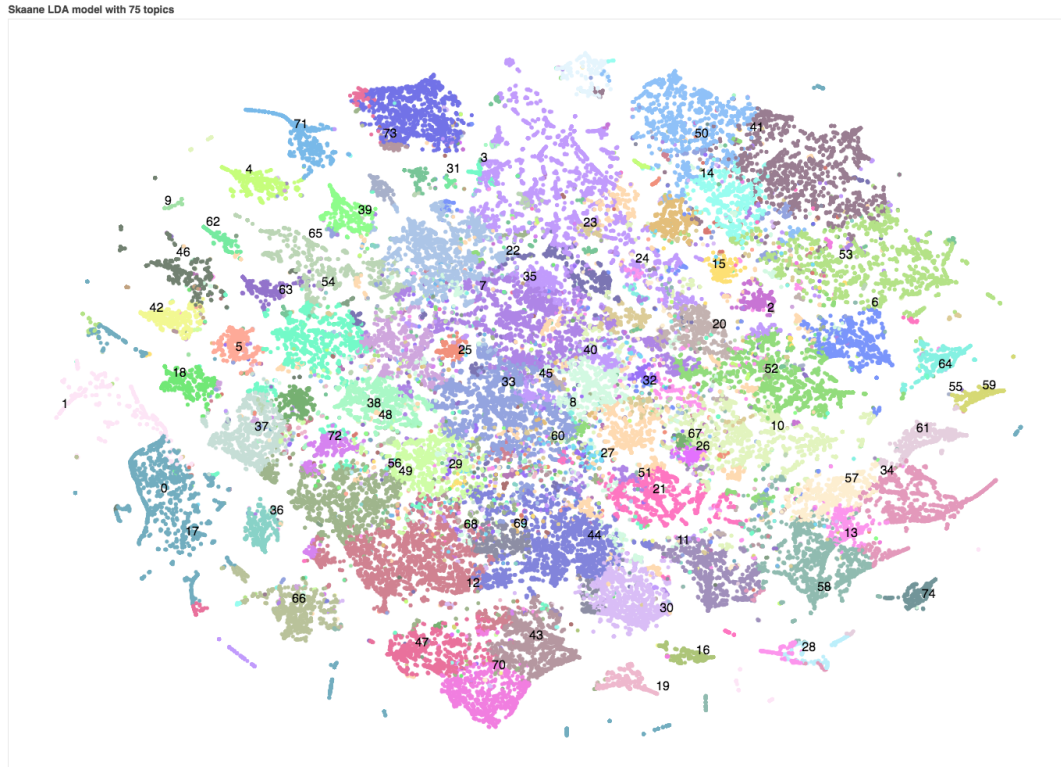


Figure 3.6: A 2D plot of the text data, where each point is colored by topic with the highest probability

exploration of the topics, the topic with the highest probability for a given data point was used to determine the color. A plot of this can be seen in Figure 3.6. Although it might be hard to interpret as a 2D plot in this report, the bokeh³ library allowed for the generation an interactive plot. Hovering over each data point would show the content of the report and the topics assigned to it, making it a convenient way to explore the data and the generated topics.

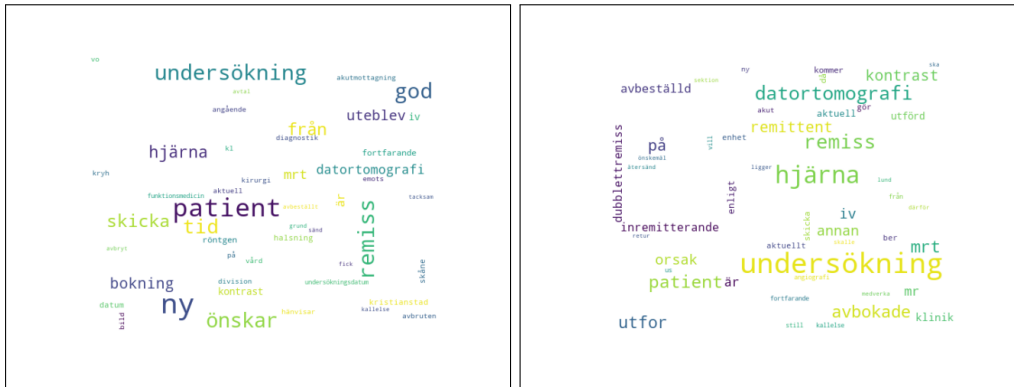
Samples of the generated topics can be seen in Figure 3.7. Another way to visualize the topics for inspection is using the LDAvis technique described by Sievert et al. [35]. They propose a *relevance* measure where the probability for a certain term within a topic is weighted against how common that topic is in the entire corpus. The interactive interface provided by pyLDAvis⁴ can be seen in Figure 3.8. This provided a good way to explore the important words in each topic.

3.3.1 The Relationship Between Topics and Categories

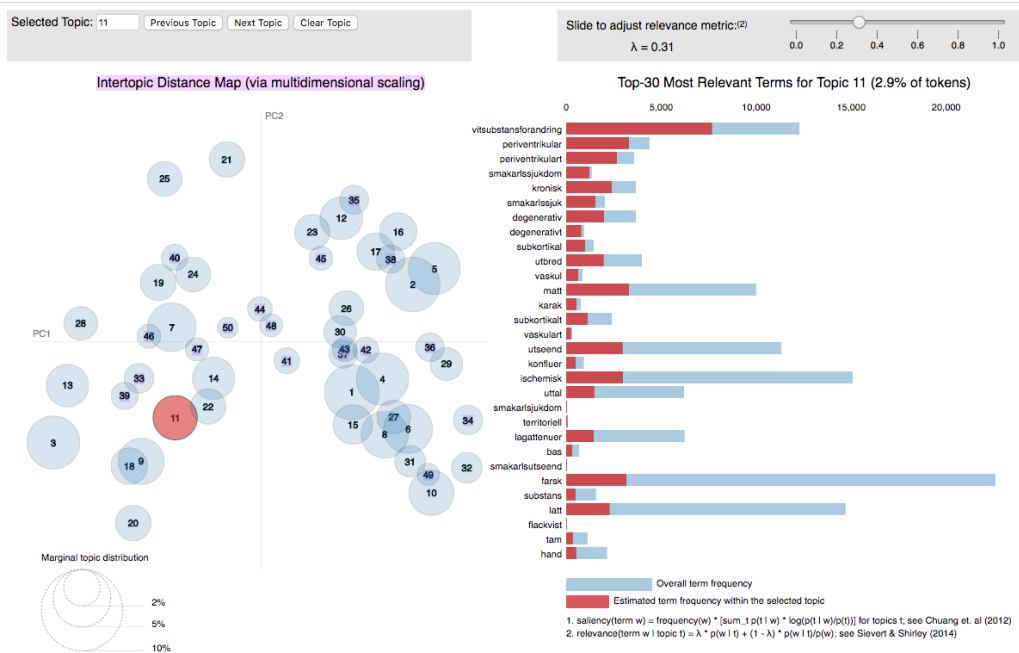
With the initial set of labeled reports, it was natural to explore and try to find a relationship between the initial set of categories and the inherit structure of the data. This was done by visualizing the LDA model again. In order to find any existing relationship between the topics and the labeled samples, the points were colored based on their assigned labels. The color of a point in the plot was based on the first label of a sorted list of the point's labels. Unlabeled samples were hidden from the plot. The resulting plot can be seen in Figure 3.9. Even if the categories are not the same in the final system, knowledge of an existing relationship might still be exploited even if the specifics change. If there is an existing relationship between categories and topics, it is likely to remain since the new categories still aim to describe the reports.

³Bokeh, <https://bokeh.pydata.org/en/latest/>

⁴pyLDAvis, <https://github.com/bmabey/pyLDAvis>



(a) A wordcloud over the words occurring in topic 1 for an 75 topic LDA model.



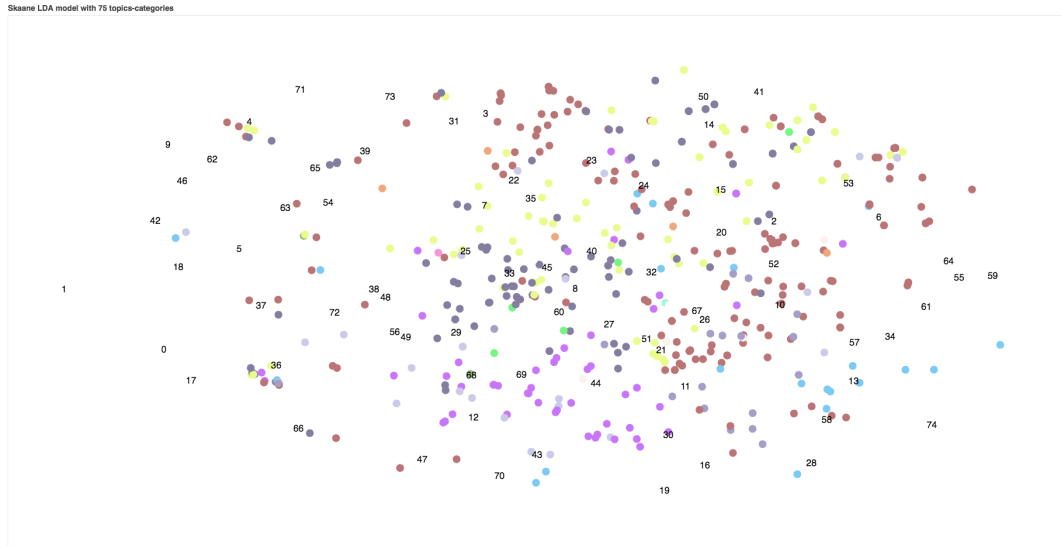


Figure 3.9: The labeled data points plotted in 2D, and colored based on the first label of the report in alphabetical order.

From this plot it is clear that there is a grouping of labels. Certain labels are more likely to occur in documents assigned a specific topic. For example, the purple points in the lower part of the graph represents the “blödning” label, the gray labels in the middle represents “infarkt” and the light blue ones that are mostly concentrated in the bottom right corner represents “infektion”. It is clear that these labels are not evenly spread out over all the topics, but neither are they confined enough to make a direct mapping between topics and labels.

To explore the relationships in more detail, they can be analyzed with the help of histograms over the topics for a certain label, as well as histograms over categories that had a certain topic as its most likely topic. This was done on the 4 most common labels, since most of the others lacked the amount of reports necessary to identify a clear pattern. In Figure 3.10 the counts of most likely topics for the four most common categories are displayed. The four most common categories were used because the others lacked the amount of reports necessary to identify a clear pattern. These categories are “infarkt”, “kärleksjukdom”, “normal” and “blödning”. From the histograms it is clear that documents of a certain category are more likely to be assigned certain topics, at least in these cases. Even though there exist a clear relationship, it is not exclusive enough to make any clear relation. The number of topics assigned and reports labeled for these 4 topics can be seen in Table 3.2.

In order to analyze these topics further, Figure 3.11 shows the different categories that has a certain topic assigned to it as the most likely one. Taking into account the information from Table 3.2, i.e. that some categories are a lot more common than others in the labeled dataset, there is not a clear enough pattern to distinguish between different categories based on the topics. This does not take the multi-label nature of the data into account. If a report has multiple labels assigned to it, both of the labels are counted separately.

3.3.2 The Relationship Between Topics and Invalid Reports

The next step was identifying the topics that were assigned to the invalid reports. First, topic 1 and 17 were identified as interesting based on the word distribution they represented. This was done using both the 2D plot of the data and LDavis visualization. The most common terms for these two topics can be seen in Figure 3.7.

As described in Section ??, a set of reports were labeled as invalid or valid in order to evaluate the technique. This set is split into a training set and a validation set. In order to

Label	No. reports	No. most likely topics
Normal	181	28
Tumör	29	15
Infarkt	105	27
Blödning	69	19
Kärlsjukdom	145	28
Hydrocefalus	10	5
Demens	14	9
Trauma	27	12
Cysta	9	7
Missbildning	3	3
Inklämmning	4	1
Infektion	29	15
Syndrom	2	2
Metabol	1	1

Table 3.2: The number of reports assigned a certain category, as well as the number of different topics assigned as the most likely one for reports with the given category.

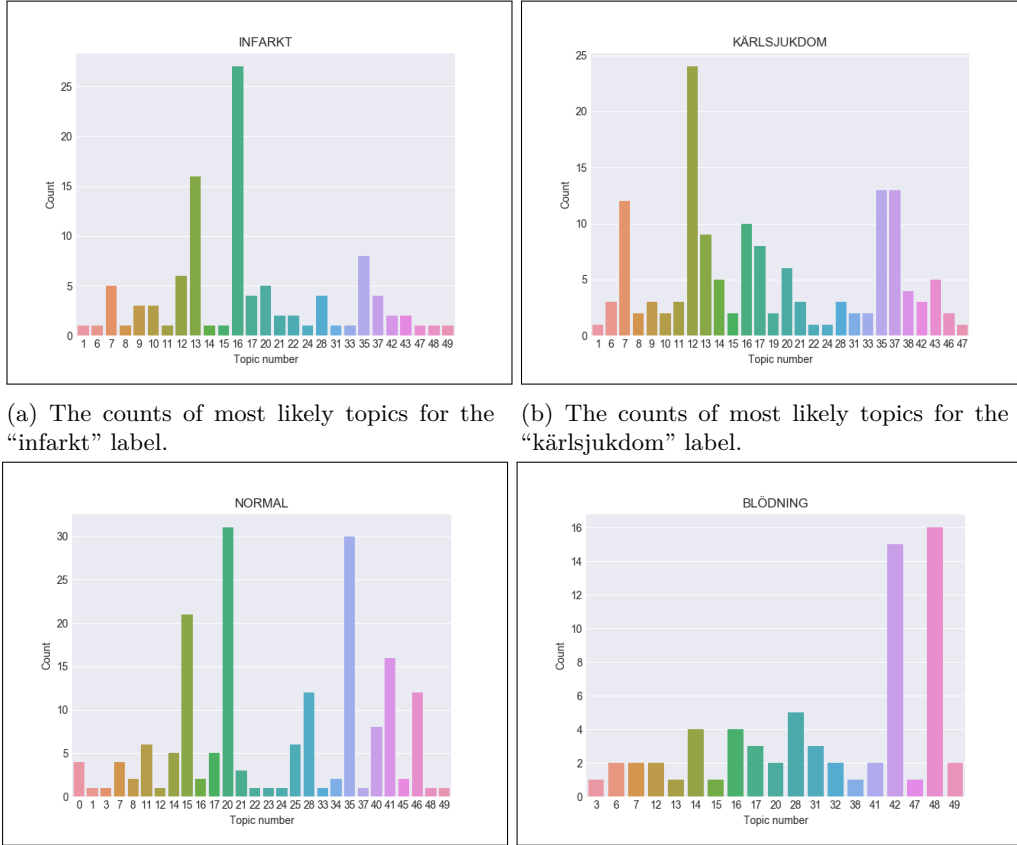


Figure 3.10: The counts of the most likely topics for the four most common categories.

verify and gain further insight how invalid reports were related to topics, the training set was used for further analysis.

The distribution of the topics with the highest likelihood for the invalid reports in the training set can be seen in Figure 3.12 (a). A couple of topics, 1 and 17 clearly stands out

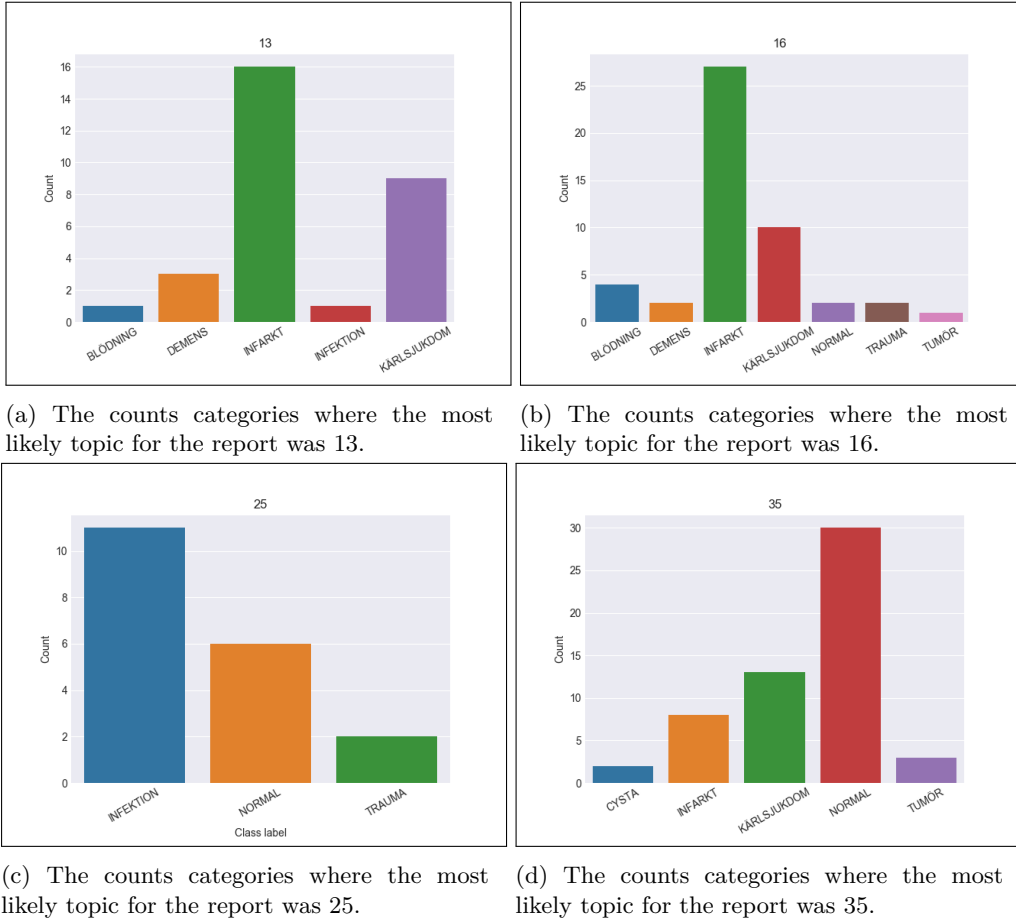


Figure 3.11: The categories of the different reports that are assigned a certain topic as the most likely one.

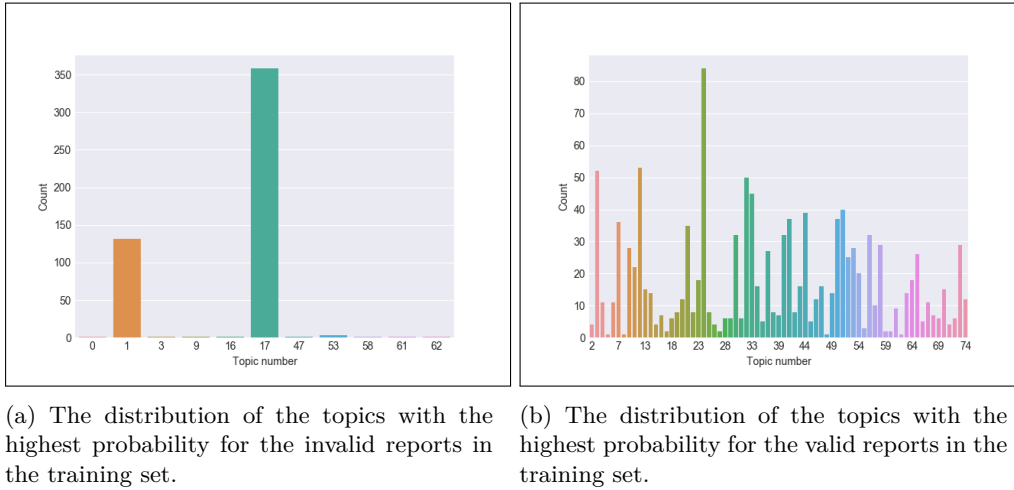


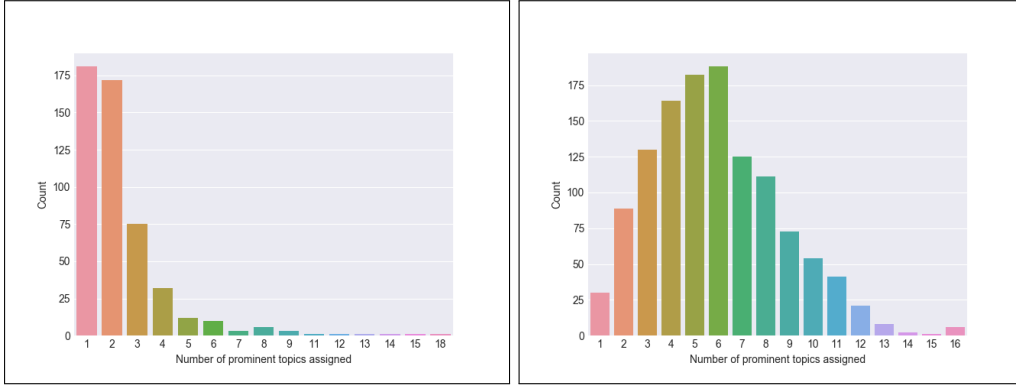
Figure 3.12: Distribution over the most likely topics for the valid and invalid reports. Note that only topics that occurred at least once are shown in the histogram.

as the ones that most invalid reports gets assigned. Specifically, topic 1 had a count of 131 reports from the invalid reports in the training set, and topic 17 had 358. Some invalid reports

have topics 0, 3, 9, 16, 47, 53, 58, 61, 62 as the most likely topic. 53 is the third most common, with a count of 3.

The corresponding plot for the valid reports can be seen in Figure 3.12 (b). There is a lot more variety among the most likely topics here. Topics 1 and 17 occur very infrequently. Topic 1 occurs 0 times, while topic 17 occurs 2 times. The third most common topic from the invalid reports, 53, occurs 28 times. The ones having topic 17 had 4 and 8 prominent topics assigned to them.

Each topic that was assigned to a report with a probability above 10% was considered to be a prominent topic. The number of prominent topics for the invalid reports can be seen in Figure 3.13 (a). 1, 2 and 3 number of prominent topics are the most common. And there are barely any reports having more than 6. The corresponding plot for the valid reports can be seen in Figure 3.13 (b). In the case of valid reports, 6 is the most common number of prominent topics. The topics are a lot more spread out than in the case of the invalid reports.



(a) The distribution of the number of prominent topics assigned to the invalid reports in the training set.

(b) The distribution of the number of prominent topics assigned to the valid reports in the training set.

Figure 3.13: The distribution of the number of prominent topics for the two categories.

Another idea was to evaluate whether or not 17 and 1 were among the most probable topics for the reports. A simple evaluation of this on the set of valid reports showed that it was not a good approach. For example, seeing if 17 or 1 had more than 10% probability returned 48 and 45 valid reports, respectively. Checking if both were above the threshold returned 11 reports.



4 Experiments

In this chapter, the experiments and their results are described. The first experiment is concerned with the evaluation of the model performance on the labeled dataset that is provided by a sampling strategy. The second one analyzes the labeled dataset that was created by the sampling strategy. For the last experiment, an analysis and evaluation of how invalid reports can be filtered out are performed.

4.1 Experiment 1: How Well do the Sampling Strategies Perform?

This experiment aimed at evaluating how well the strategies perform, by looking into how they effect the SVM model, according to certain metrics. The results of this experiment is used to answer the first research question.

4.1.1 Method

The active learning strategies that were evaluated are:

- *BSVM (BSVM)*: Described in Section 2.2.3
- *Maximum Loss Reduction with Maximum Confidence (MMC)*: Described in Section 2.2.4
- *Adaptive Active Learning (AAL)*: Described in Section 2.2.5

The motivation behind the choice of strategies will be discussed in Chapter 5.

In order to provide a thorough evaluation of how well they techniques perform a set of already labeled documents was needed. For this reason, the Reuters-21578 dataset was used, as is discussed in Section 3.1. The properties of the dataset, as well as a comparison between it and the clinical data provided by Sectra can be found in Section 3.1. The dataset is common in active learning research and has been used by Brinker et al. [9] and Yang et al. [42], among others. With this dataset, the same pre-processing steps that were applied to the clinical dataset were applied to the Reuters data too. Some modifications of this includes the stopwords, instead of a curated list of words, the unmodified list of english stopwords provided by nltk was used. The main goal was to compare how the different techniques affected the labeled dataset, and how well an SVM model performed on it. Optimizing the process for the particular model and dataset was therefore not the focus of the study, but instead offering a

ID	Active Learning Strategy	Initial Sampling	Initial Sample Size
1	BSVM	Random	25
2	BSVM	Random	50
3	BSVM	Random	100
4	BSVM	Sampled from clusters	25
5	BSVM	Sampled from clusters	50
6	BSVM	Sampled from clusters	100
7	MMC	Random	25
8	MMC	Random	50
9	MMC	Random	100
10	MMC	Sampled from clusters	25
11	MMC	Sampled from clusters	50
12	MMC	Sampled from clusters	100
13	AAL	Random	25
14	AAL	Random	50
15	AAL	Random	100
16	AAL	Sampled from clusters	25
17	AAL	Sampled from clusters	50
18	AAL	Sampled from clusters	100

Table 4.1: The different configurations of active learning strategies evaluated.

more comprehensive comparison. With this set of labeled reports, a simulation of the labeling process was run.

The strategies need a small initial set of labeled reports which they can use to get initial predictions from the SVM model, upon which the strategies base their calculations. The techniques were evaluated both by selecting this initial set of points at random, as well as selecting them from the clusters generated by the k-means algorithm. Sampling from the clusters was done by iterating over the clusters and selecting an equal number of data points from each clusters. All samples selected from a given cluster were chosen randomly amongst the members of the clusters. Furthermore, the number of clusters selected was 25, in order to get an equal number of reports from each cluster in the different experimental settings. Topic vectors from the topic model in Section ?? was used as input to the k-means algorithm.

Since the different models may depend on the initial samples in different ways, different initial sizes were evaluated. This is also done by Yang et al. [42]. In their paper they tried quite large initial sample sizes. Here, the sizes evaluated are: 25, 50, 100. The reason for this is that an large initial sample size would make it hard for the human annotator to see a difference in the class balance early on. In each iteration, 25 labels were queried. The ones selected every iterations were the 25 best one according to the strategy’s measure. 100 iterations were run per strategy. Making it 2500 labels that are labeled, in addition to the initial sample. The different active learning configurations that were tried is displayed in Table 4.1 In total there were 18 configurations, based upon the three different methods. Every configuration was evaluated 5 times, and it is the average of those runs that are considered to be the final results. The reasoning behind this is to reduce the effect of a single random selection.

In each iteration, the SVM model was evaluated by being trained on the labeled data, and evaluated on the test set described in Section 3.1. How good the SVM model’s predictions are were evaluated with the following metrics:

1. Accuracy
2. Micro recall
3. Macro recall
4. Micro precision

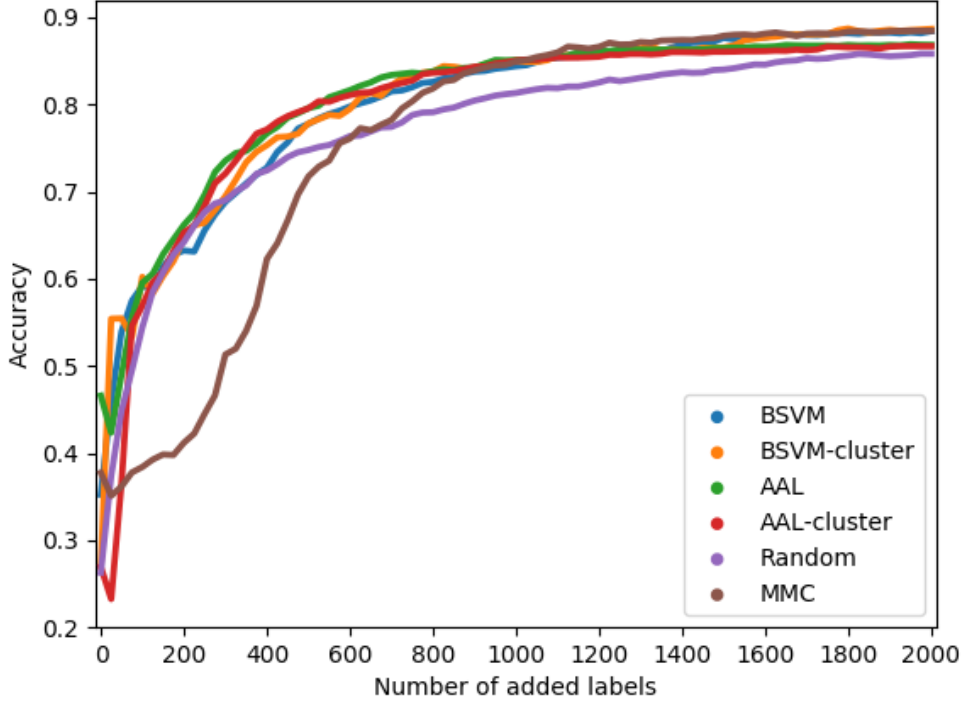


Figure 4.1: Accuracy of the models with initial sample size 25

5. Macro precision
6. Micro F1-Score
7. Macro F1-Score

These are described in Section 2.4 and are frequently used to compare different active learning methods, for example by Yang et al. [42], Dasgupta et al. [12] and Li et al. [23]. In addition to this, how long time they take to run is compared as well.

The method here should be rather easy to replicate. Metrics are clearly defined, and the configurations used for the different strategies are laid out.

4.1.2 Results

The first result is the SVM model’s accuracy. Its evaluation when trained on data provided by the active learning strategies with initial sample sizes of 25, 50 and 100 can be seen in Figure 4.1, Figure 4.2, and Figure 4.3 respectively. Note that when retrieving the initial sample from the clusters, only samples with label cardinality 1 was retrieved after 10 tries. MMC requires at least two different label cardinalities in the initial sample in order for the logistic regression model to work. For that reason, MMC with cluster initialization was not evaluated. In Table 4.2 to Table 4.4 it can be seen how many labels were required to reach a certain accuracy.

The micro and macro F_1 -score, recall and precision for the initial sample size of 25 can be seen in Figure 4.4. The same evaluation for the initial sample size of 50 and 100 can be seen in Figure 4.5 and Figure 4.6, respectively.

A comparison of how long time it took for the different strategies can be seen in Figure 4.7.

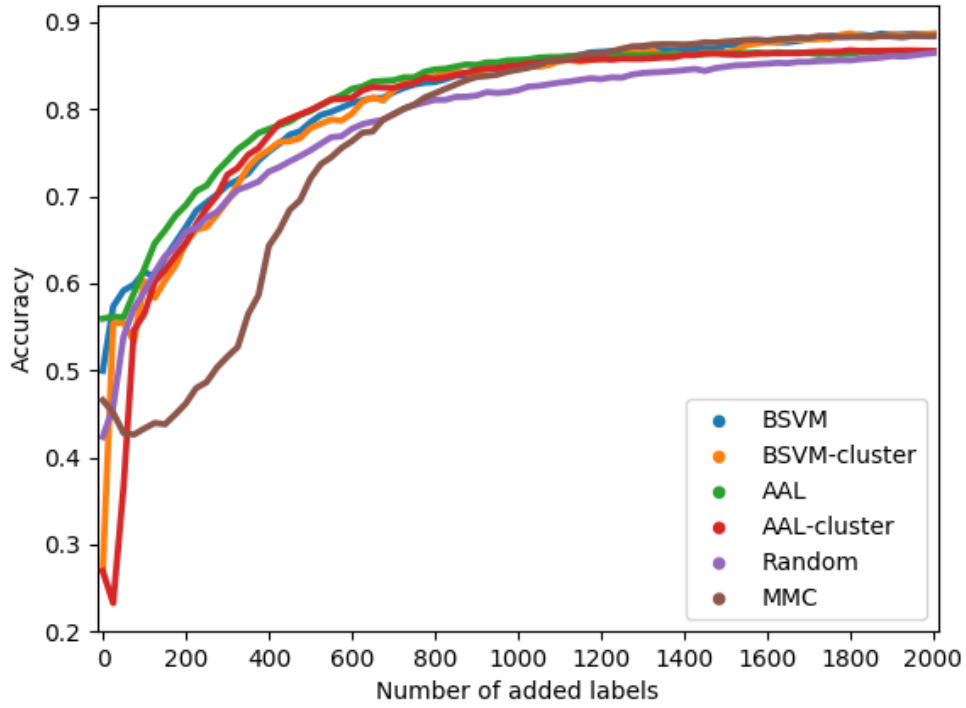


Figure 4.2: Accuracy of the models with initial sample size 50

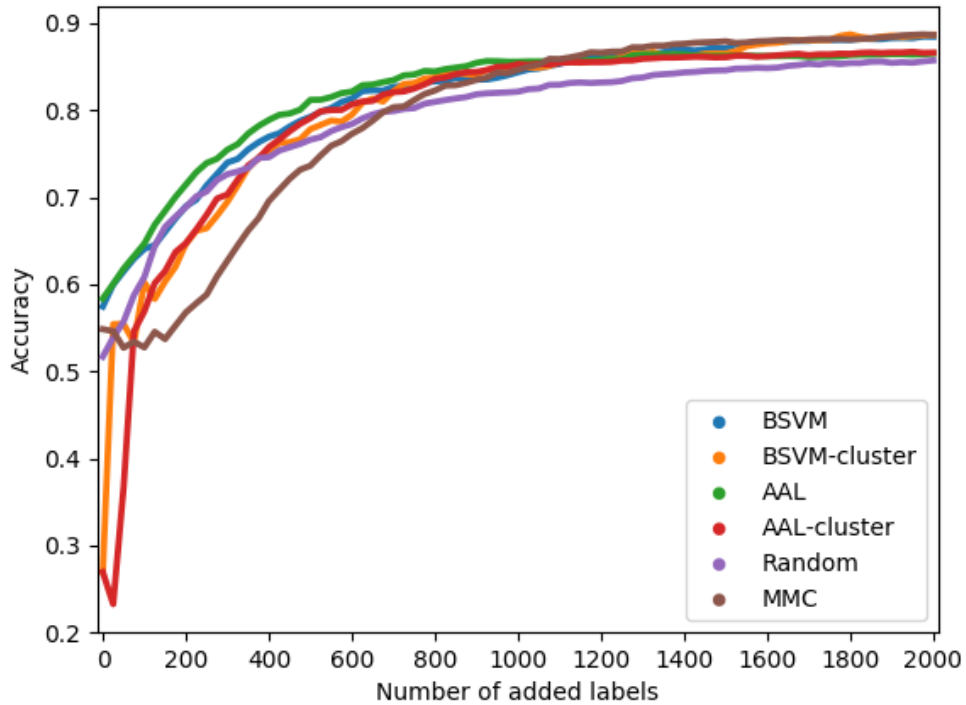


Figure 4.3: Accuracy of the models with initial sample size 100

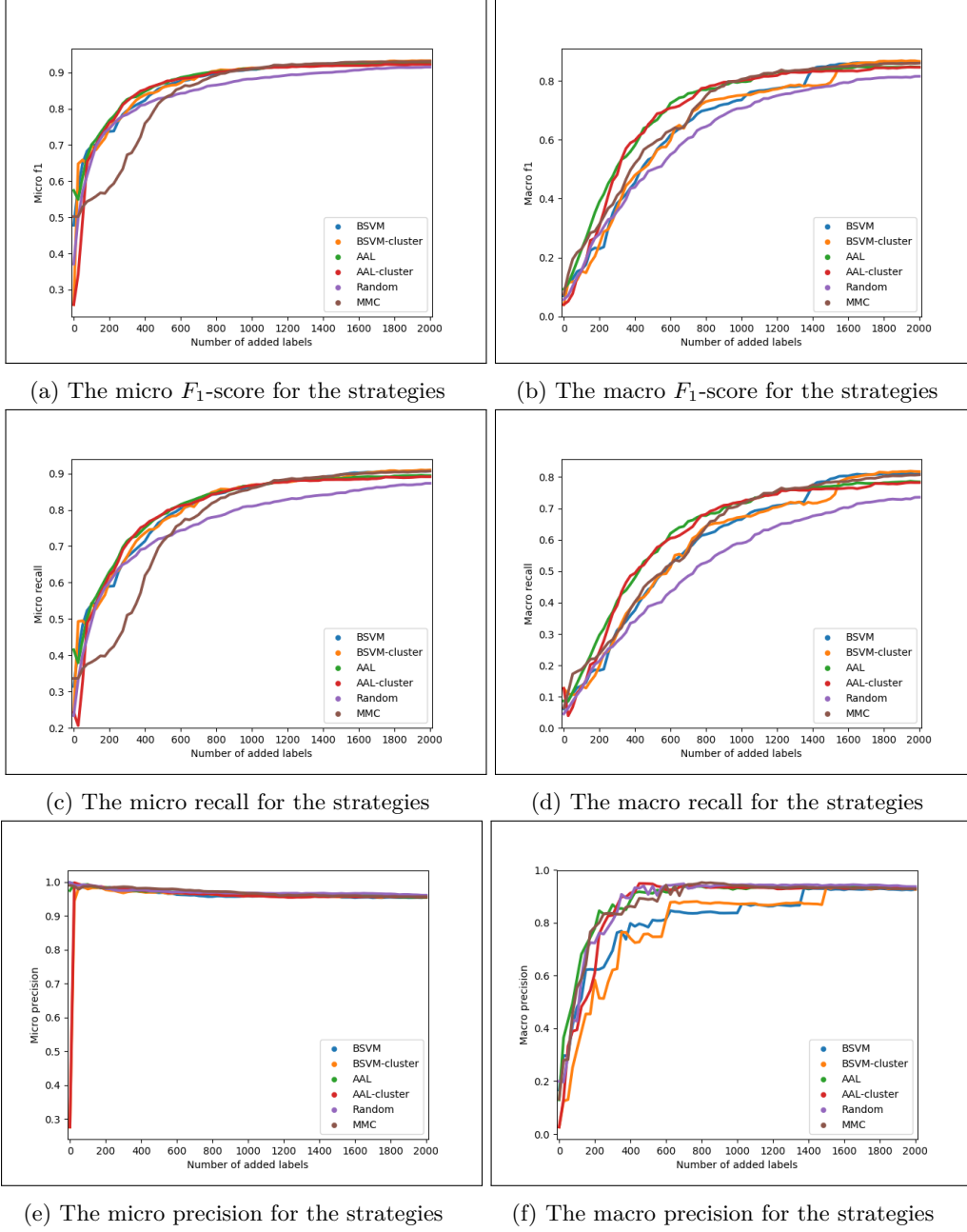


Figure 4.4: The metrics for the strategies when the initial sample size of 25 was used.

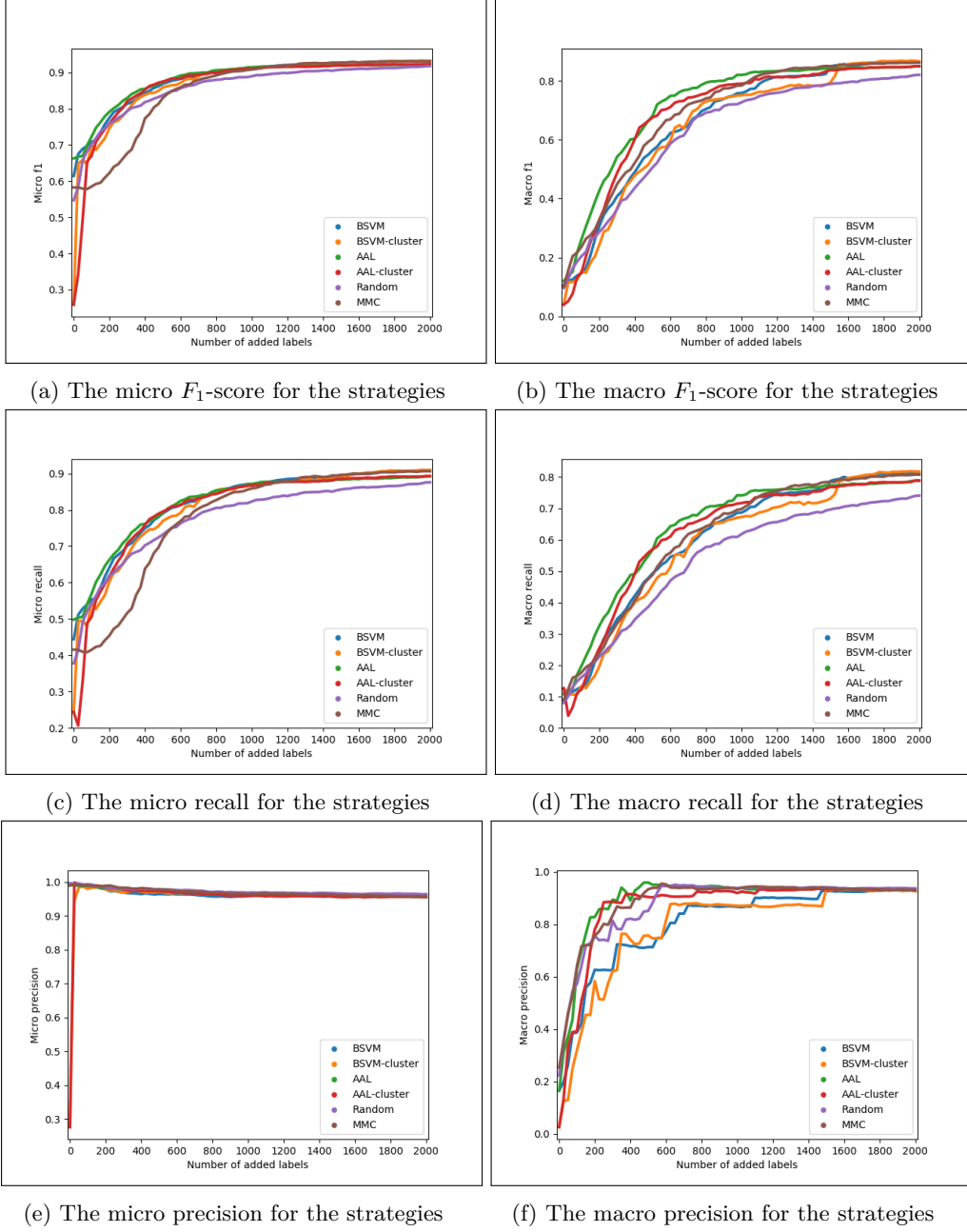


Figure 4.5: The metrics for the strategies when the initial sample size of 50 was used.

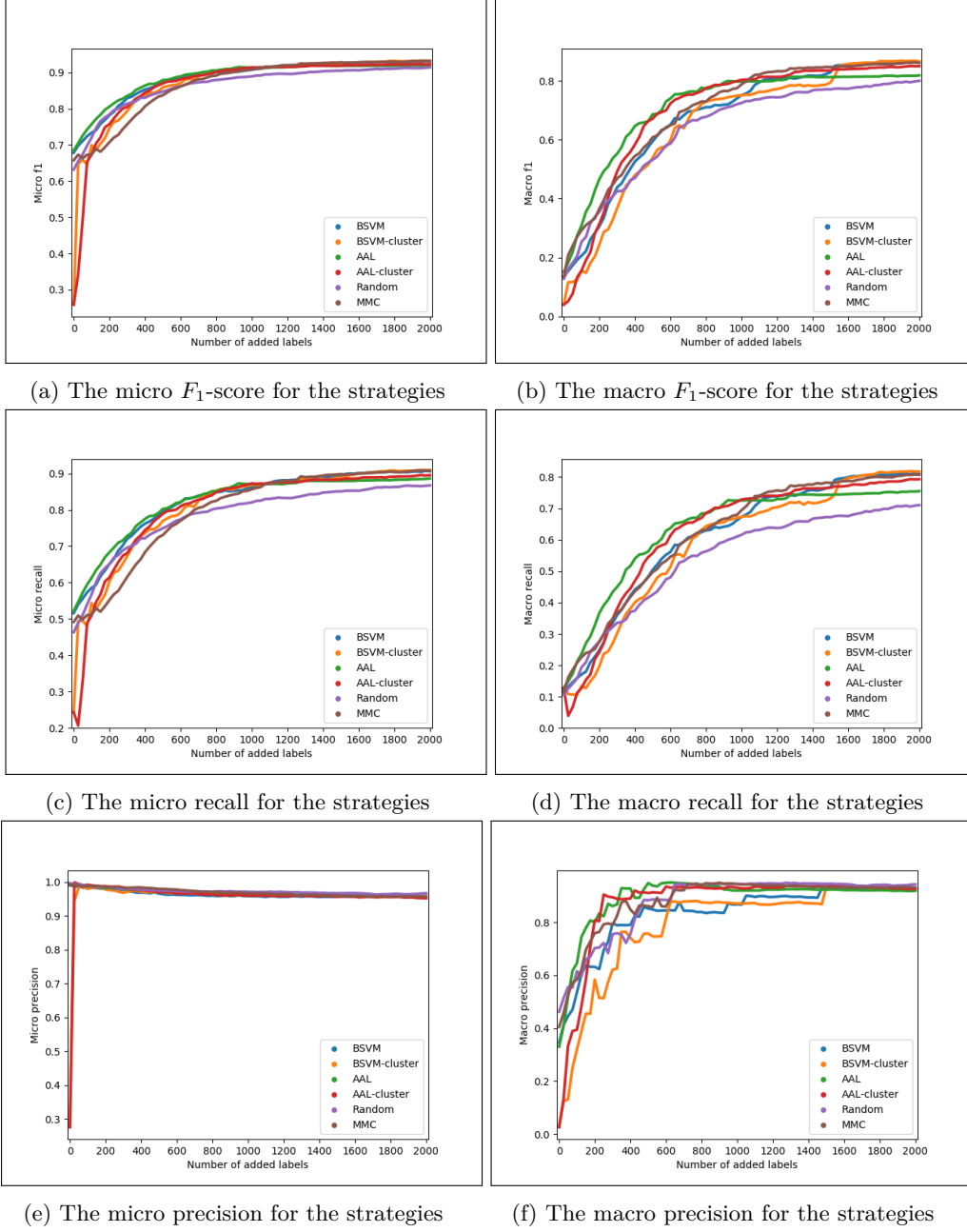


Figure 4.6: The metrics for the strategies when the initial sample size of 100 was used.

Strategy	Initial Sample	75 %	80 %	85 %	87 %	88 %	89 %
BSVM	Random	475	650	1100	1450	1675	2425
BSVM	Cluster	425	650	1100	1575	1700	2425
MMC	Random	600	775	1050	1250	1575	N/A
MMC	Cluster	N/A	N/A	N/A	N/A	N/A	N/A
AAL	Random	400	575	975	2425	N/A	N/A
AAL	Cluster	375	550	1025	2300	N/A	N/A
Random	Random	550	900	1700	N/A	N/A	N/A

Table 4.2: The number of labeled reports in total that the strategies required to achieve the different accuracy values, with initial sample size 25. Only results for the first 2500 data points that were labeled are considered.

Strategy	Initial Sample	75 %	80 %	85 %	87 %	88 %	89 %
BSVM	Random	450	625	1075	1550	1750	N/A
BSVM	Cluster	450	675	1125	1600	1725	2450
MMC	Random	625	775	1100	1325	1675	N/A
MMC	Cluster	N/A	N/A	N/A	N/A	N/A	N/A
AAL	Random	375	575	925	N/A	N/A	N/A
AAL	Cluster	425	575	1075	N/A	N/A	N/A
Random	Random	550	775	1575	N/A	N/A	N/A

Table 4.3: The number of labeled reports in total that the strategies required to achieve the different accuracy values, with initial sample size 50. Only results for the first 2500 data points that were labeled are considered.

Strategy	Initial Sample	75 %	80 %	85 %	87 %	88 %	89 %
BSVM	Random	400	600	1125	1525	1775	N/A
BSVM	Cluster	450	675	1125	1600	1725	2450
MMC	Random	600	750	1100	1325	1675	N/A
MMC	Cluster	N/A	N/A	N/A	N/A	N/A	N/A
AAL	Random	350	525	925	2450	N/A	N/A
AAL	Cluster	450	600	1025	N/A	N/A	N/A
Random	Random	475	775	1700	N/A	N/A	N/A

Table 4.4: The number of labeled reports in total that the strategies required to achieve the different accuracy values, with initial sample size 100. Only results for the first 2500 data points that were labeled are considered.

4.2 Experiment 2: Evaluating the Label Balance on the Labeled Dataset

The goal with the second experiment was to evaluate how the labels in the produced labeled dataset are distributed. A set where the labels are more evenly, or uniform, distributed would be preferable. From a perspective of the person labeling, it could feel more productive not assigning the same labels most of the time. The more prosperous outcome from a balanced dataset would be that the models using the data in later stages could also benefit from this, and obtain better results. This experiment is used to answer the second research question.

4.2.1 Method

The configurations used here are the same as in Table 4.1. Every iteration, the distribution of labels assigned are stored and analyzed. He et al. discusses the usage of ROC curves and measures such as g-means to compare multi-class imbalanced data [15]. However, the doctor

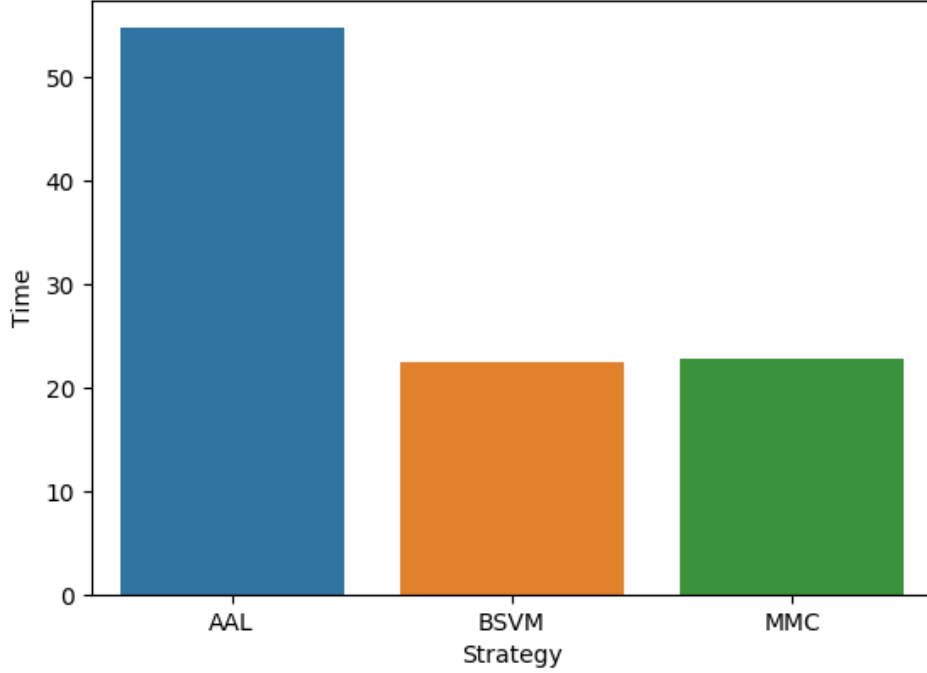


Figure 4.7: The percentage of time used on the different strategies during one iteration.

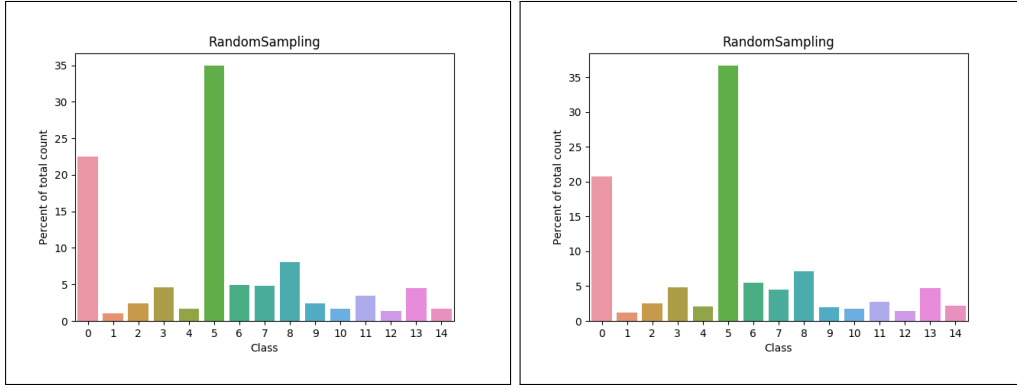
involved at Sectra specifically requested that labels should be more uniformly distributed. The evaluation will therefore focus on measuring that instead of the models performance, which is done in ???. An evaluation of how the distribution progresses was done by comparing how the class imbalance is affected by the number of new samples obtained from the different methods.

Ertekin et al. [13] used a class imbalance ratio to compare how well an active learning strategy worked. This was only done for the binary case. In order to get a measure for the multi-label problem, the evaluation of class imbalance was measured by the percentage of all total labels that were in the most common class, as well as in the top 3 most common classes. Furthermore, the ratio between the biggest and the smallest class will be used in the evaluation.

This approach is also replicable. The configurations are clearly defined, and so are the metrics used. The validity of the experiment is discussed in Section ???.

4.2.2 Results

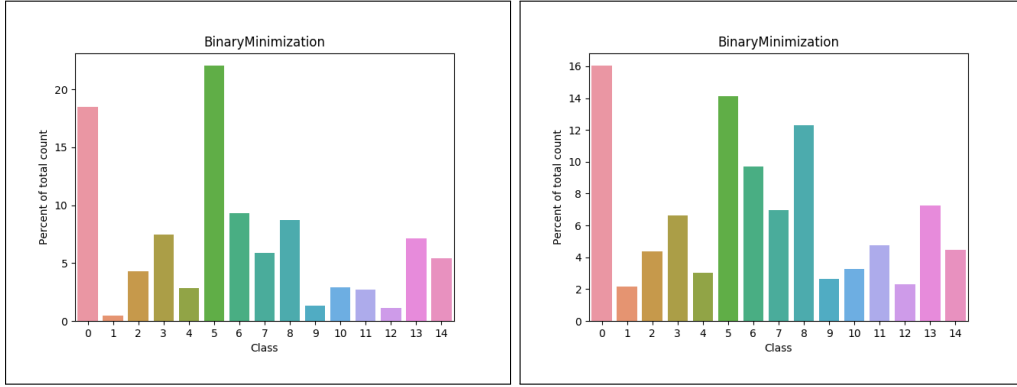
The results of how the different strategies affected the balance of the labeled dataset is now presented. For the Reuters dataset, how the overall distribution of the labels is can be seen in Figure 3.3. After random sampling, the distribution can be seen in Figure 4.8. For comparison with the techniques it shows the labels both after 500 labels are added, and 2000. In order to be able to compare it with the other techniques easily, the plot contains the distribution after both 500 and 2000 labels are acquired. The distribution after sampling with the original BSVM can be seen in Figure 4.9, and with the initial samples taken from clusters in Figure 4.10. For MMC the distribution can be seen in Figure 4.11. The corresponding plots for Adaptive Active Learning can be seen in Figure 4.12 and Figure 4.13.



(a) The class distribution from random sampling after 500 labels

(b) The class distribution from random sampling after 2000 labels

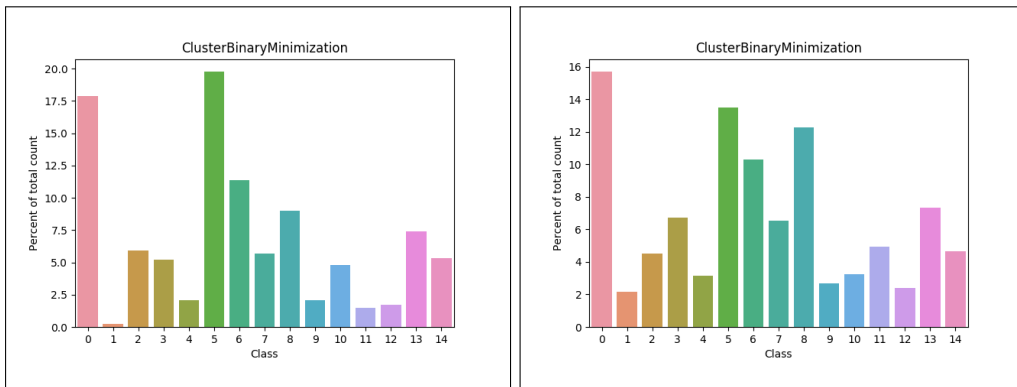
Figure 4.8: The distribution of labels after random sampling



(a) The class distribution from BSV after 500 labels

(b) The class distribution from BSV after 2000 labels

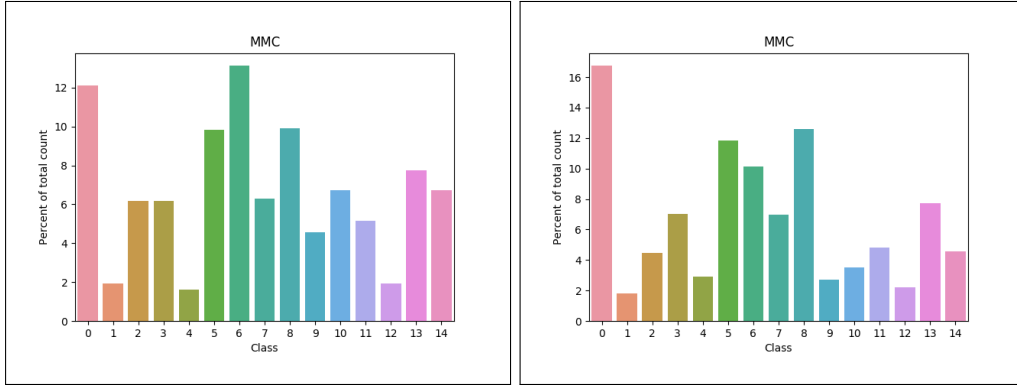
Figure 4.9: The distribution of labels after BSV



(a) The class distribution from BSV, with the initial sample from clusters, after 500 labels

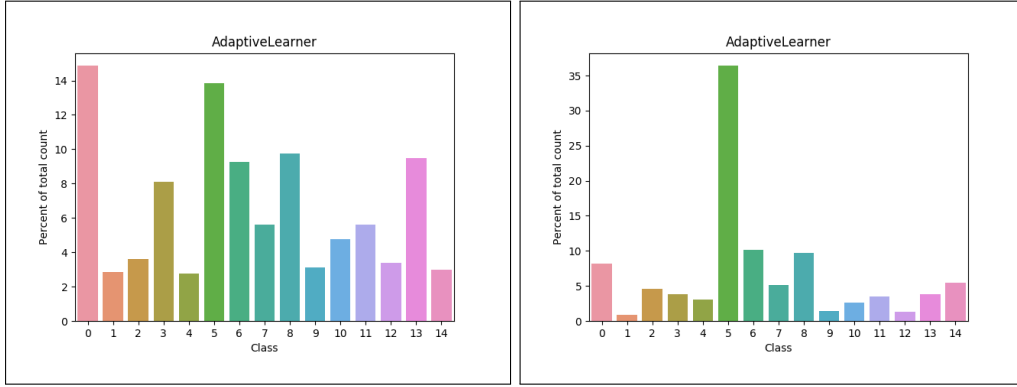
(b) The class distribution from BSV, with the initial sample from clusters, after 2000 labels

Figure 4.10: The distribution of labels after BSV with clustering



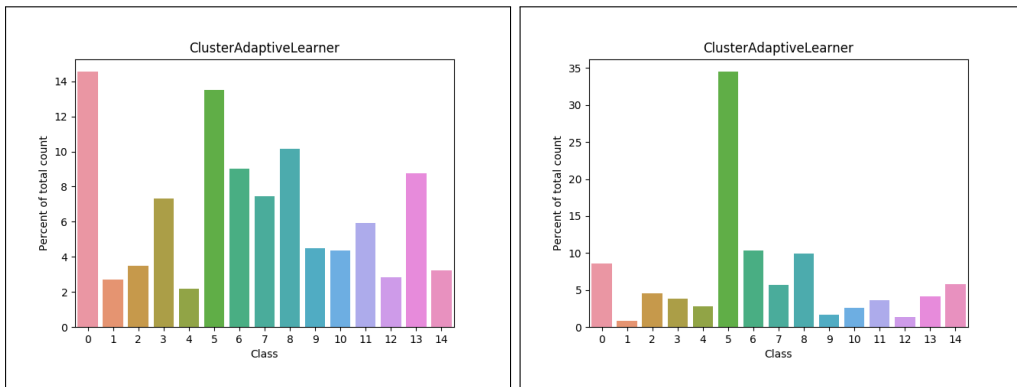
(a) The class distribution from MMC after 500 labels (b) The class distribution from MMC after 2000 labels

Figure 4.11: The distribution of labels after MMC



(a) The class distribution from Adaptive Active Learning after 500 labels (b) The class distribution from Adaptive Active Learning after 2000 labels

Figure 4.12: The distribution of labels after Adaptive Active Learning



(a) The class distribution from Adaptive Active Learning, with the initial sample from clusters, after 500 labels (b) The class distribution from Adaptive Active Learning, with the initial sample from clusters, after 2000 labels

Figure 4.13: The distribution of labels after Adaptive Active Learning with clustering

In Table 4.5 the results of the evaluation after 500 new labels can be seen. The corresponding table for the evaluation after 2000 new labels can be seen in Table 4.6.

Strategy	Initial Sample	Top Class	Top 3 Classes	Small/Big Ratio
Random	Random	34.9 %	65.5 %	32.0
BSVM	Random	22.0 %	49.8 %	45.0
BSVM	Clusters	19.8 %	49.0 %	80.0
Adaptive	Random	14.8 %	38.5 %	5.4
Adaptive	Clusters	14.5 %	38.2 %	6.65
MMC	Random	13.1 %	35.2 %	8.2

Table 4.5: The results after analyzing the label distribution after 500 new labels has been added.

Strategy	Initial Sample	Top Class	Top 3 Classes	Small/Big Ratio
Random	Random	36.65 %	64.6 %	29.4
BSVM	Random	16.0 %	42.4 %	7.5
BSVM	Clusters	15.7 %	41.5 %	7.3
Adaptive	Random	36.4 %	56.2 %	43.4
Adaptive	Clusters	34.5 %	54.8 %	41.0
MMC	Random	16.7 %	41.2 %	9.3

Table 4.6: The results after analyzing the label distribution after 2000 labels has been added.

4.3 Experiment 3: Filtering out Invalid Reports

This experiment is concerned with the identification of invalid reports. The task here was to evaluate how well unsupervised techniques could be used to filter out these invalid reports. Invalid reports are considered to be reports that describe a situation where an examination never took place. This can be because of a deceased patient, a patient being moved to another hospital, a patient did not show up or for some reason did not want to go through with the examination. The results of this is the base for answering the third research question.

4.3.1 Method

In order to evaluate how well the selected model performed on the clinical data, a set of reports had to be marked as valid/invalid. This was done by creating a script that presented a report to the user, and requested the label. 5358 reports were labeled by the author, out of which 623 were marked as invalid. The labels were skewed, containing only 623 invalid reports, which makes up 10.4% of the labels. In order to obtain a more balanced dataset, 3653 of the valid reports were dismissed at random making it 1000 in total. In order to make the models able to separate the invalid reports from the valid reports they had to be manually analyzed. They are both unsupervised methods and were therefore not fitted with a specific target. Approaching this in a way that would not result in the approach being overfitted to the analyzed data could potentially be hard since they are manually analyzed. To reduce the bias in the evaluation, the labeled reports were split into a training set and validation set, containing 80% and 20% of the reports respectively.

The training set was further used to be verify the topics that were identified as important, whereas the validation set was used to evaluate how well the approach performed. This manual identification can be seen in Section 3.3.2. Based on these findings, reports were determined to be invalid or not based on if they fulfilled both of the following criteria:

- Having either topic 1 or 17 as its most probable topic.

- Not having more than 6 prominent topics assigned to it.

Where prominent topics are defined as those who has a probability of more than 10% for a given report.

After the initial exploration it was clear that the topic vectors contained patterns ripe for exploitation. A topic vector is the vector with the topic probabilities for a document. The patterns are clear enough to motivate the manual identification of topics that are important to differentiate between invalid and valid reports. In order to compare this result with some more objective baseline, a logistic regression classifier was fitted on the data and evaluated as well. A set of reports were already labeled with “invalid” or “valid” in order to evaluate the manual interpretation approach. This set was thus used to fit the classifier. The topic vectors were used as features, and the targets were the labels indicating if a report is valid or not.

4.3.2 Results

The evaluation on the validation set can be seen in Table 4.7. In the figure you can also see the results of the logistic regression classifier, which was fitted with the topic vectors as features and the invalid/valid labels as targets.

	Manual topic identification	Logistic regression
Precision	97.2%	98.7%
Recall	100%	100%
F_1-measure	98.6%	99.4%
Accuracy	97.9%	99.1%

Table 4.7: The results of the classification of the invalid reports. The manual identification column represents the use of manual interpretation of the LDA topics to find the invalid reports.

Both of these approaches were evaluated using four different metrics, recall, precision, F_1 -measure and accuracy. All of which are described in Section 2.4.

The replicability of this experiment is varied. It contains a manual analysis of LDA generated topics, on a proprietary dataset, which may be hard to replicate. However, performing logistic regression on topic vectors generated with the specific LDA is easier to reproduce.

4.4 Frameworks, Tools and Implementation

The entire system was written in Python. The motivation behind this choice was mainly that, when it comes to machine learning and text mining, most of the existing infrastructure at Sectra is using Python. This, in combination with the fact that there exists several tools for these purposes in Python, such as *numpy*¹, *nlTK*², *scikit-learn*³ and *gensim*⁴. Most of the plotting was done using the *seaborn*⁵ and *bokeh*⁶ libraries. *pyLDavis*⁷ was used for some additional visualization purposes with regards to topic models.

However, when it comes to active learning, there does not seem to be a proven mainstream library that contains a set of readily available algorithms. In order to achieve better integration between the active learning system and the existing infrastructure at Sectra, as well as making adaptations such as the number of items queried in each iteration, an active learning framework

¹Numpy, <http://www.numpy.org/>

²Natural Language Toolkit, <https://www.nltk.org/>

³scikit-learn, <http://scikit-learn.org/stable/>

⁴Gensim, <https://radimrehurek.com/gensim/>

⁵Seaborn, <https://seaborn.pydata.org/>

⁶Bokeh, <https://bokeh.pydata.org/en/latest/>

⁷pyLDavis, <https://github.com/bmabey/pyLDavis>

was written from scratch. The ground for this framework were the algorithms presented in Section 2.2.

This framework consisted of three modules, called *model*, *dataset*, and *query strategy*. The model is a wrapper around different machine learning models. By providing an interface for a distance or certainty measure, any underlying model able to provide such an interface can be incorporated. For accessing the data pool a dataset wrapper was written, with an interface for accessing the labeled and unlabeled pools. Putting this in its own module opens up the possibility for using several different storage solutions, such as a database or plain text files. The query strategy module contains the different active learning algorithms for selecting what sample to label next.



5 Discussion

The discussion chapter is separated into three parts. First, it goes through and analyzes the results from Chapter ???. This is followed by an analysis of the methods described in Chapter ??. At last the work is discussed in a wider context, based on the ethical and societal impact techniques like these may have.

5.1 Results

Obtaining 97.9 % and 99.1 % accuracy for classifying invalid reports must be considered a good result. The reports describing the cases where an examination did not happen uses a fairly different vocabulary. For these cases there is a lack of medical terms, while expressions such as “canceled”, or “new time” are a lot more common. This distinction makes it rather easy for the topic model to identify the topics relating to this. The vocabulary is quite unique. There is some overlap of course, the manual identification of topics did get a worse precision than the logistic regression model. That is, it got more invalid reports when trying to identify the valid ones. A probable reason for this is that there exists some information in the topics other than topic 1 and 17 that can help in identifying reports that have these as their most likely topic, but is valid. One such thing could be a more finely tuned notion of prominent topics. However, given the clear relationship between certain topics and categories, it might even be sufficient to have a well curated list of keywords to look for when classifying the reports.

Looking at how the different strategies perform in terms of accuracy and F_1 -score, it is clear that for the first 600-700 labels, MMC performs considerably worse than the rest. That is including random sampling. It is equally clear that it performs better when the initial sample is bigger. One reason behind this may be MMC’s dependence on the label cardinality of the samples. MMC’s dependence on label cardinality can be seen in Section 2.2.4. If the information on label cardinality is not varied, the predictions that MMC base its calculations upon will not be as good. In the paper by Yang et al. [42], MMC performs significantly better than Binary Version Space Minimization (BSVM) at all stages. One obvious reason for this discrepancy could be the implementation used in this paper, as is discussed in Section 5.1. Another one can be the initial sample sizes used in their paper, even when they compare different initial sample sizes, they start at 100 samples and go up to a 1000.

Adaptive Active Learning (AAL) performs the best of the evaluated strategies in both accuracy and F_1 -score. A probable reason for this is the extra computations the strategy

does in order to find the best weight between the certainty based, and the cardinality based, strategies. If the cardinality predictions is not deemed to be very good in the beginning, it would simply put more weight on the certainty measure, and vice versa.

BSVM on the other hand performs very similarly to random sampling for the first samples. The reason for this is rather simple. The strategy finds the class that is the closest to the decision boundary for each data point, and selects the minimum distance of those. If there are several documents with the same minimum value they are selected at random from those. Until all classes has been sampled at least one, the minimum for all data points will be the same class, and it will be equal among all data points. Therefore, it will be the same as selecting randomly among all the samples. However, when all classes have been sampled, and quickly outperforms the random sampling in terms of accuracy and F_1 -score.

In the end, the different strategies all perform better than random sampling. The ones that are based on MMC and BSVM tend to perform a bit better than AAL when more than 1200-1300 samples are added. This may very well be related to the fact that the distributions for MMC and BSVM is considerably more uniform when a lot of labels have been requested. BSVM is the only technique that reached 89 % accuracy within the first 2500 labels to be labeled. It is possible that other techniques would have reached it given more labels, but it is clear that BSVM reaches a peak accuracy earlier. The technique always tries to sample from the most uncertain category. The lack of reports in this category can be the thing that prohibits the other strategies from achieving a higher accuracy.

The initial clustering of reports did not seem to make as much of a difference as the strategy used. This makes sense, the initial sample size only makes up for a little bit of the overall set of labeled reports. Worth noting is that the SVM model performs worse for the first couple of iterations with initial clustering. It quickly recovers after this and seems to perform as well, or slightly better, compared to the randomly initialized counterpart. The reason for the low initial performance might be due to skewed dataset. If we assume that the clusters captures a bit of the label information, then the labels should be more evenly distributed in the initial sample. On the other hand, if the test set contains mostly one or two labels, the initial set will not have seen as much of these and will therefore not be as accurate in predicting them. This would then greatly reduce the evaluation.

All clusters will not have the same number of members. If most of the documents with a high label cardinality got assigned the same populous cluster, they might get neglected from the initial sample since an equal number of reports was taken from each cluster. This could be a reason for the clustering approach to initial sampling only obtained samples with label cardinality 1, and therefore excluding MMC from the process.

Looking at the distribution of the labels it is instantly clear that the three strategies outperforms random sampling significantly. MMC and AAL are by far the most balanced one after only 500 labels are labeled. When 2000 data points are labeled, BSVM is the most balanced one, albeit slightly compared to MMC. It can probably be attributed to it selecting the most uncertain class in every iteration. The most uncertain class is probably the one with the fewest samples, causing the distribution to be more uniform over time.

While BSVM always selects the sample that is the most uncertain in one single class, one can view MMC and AAL as more of an average of the uncertainty over the categories for a data point. They also incorporate the label cardinality information. This averaging trait may be beneficial for the distribution early on. The model is most likely rather uncertain about most categories, so instead of focusing on the one with the absolutely lowest value they select the data point that will give the overall most information. As with the evaluation on model performance, BSVM seem to perform better in the long run. One could include MMC in this too, the difference between the two could probably be neglected in most applications. However, if only a few hundred labels are supposed to be labeled, AAL is unrivaled when all metrics are considered.

5.2 Method

Results that are based on a public dataset are naturally easier to reproduce. The method becomes more reliable in the sense that the same results can be expected by reproducing the concrete steps. However, the clinical dataset from Sectra is not publicly available. Thus any results that are derived from specific attributes of that dataset might not be exactly reproducible in a new environment. The comparisons of the active learning algorithms are using the Reuters dataset, which is both public and a standard dataset for evaluation. Reproducing these results might therefore be more reasonable.

During the exploration phase and the first experiment, the study of the LDA model and its topics was in some part based on the author's intuition. For this reason, if another party would perform the same study they might identify other patterns. For example, the 10% threshold put on what is considered a prominent topic was purely based on intuition after exploring the dataset. However, the patterns are later studied in a more objective way when they are visualized in the form of relationships between the topics and assigned labels. The manual identification of topics in the first experiment is then compared to a more objective solution in the form of the logistic regression classifier. Results derived from this classifier can therefore be seen as more reliable and reproducible. Any subsequent study is more likely to obtain similar results this approach.

Another aspect of the experiment on invalid reports is the labeling process. This was done manually by the author, without any medical knowledge. Some reports may have been misidentified, but the nature of the labeling is rather trivial in this case. The medical knowledge required to understand the result of an examination is far greater than the one needed to see if an examination was performed. Which in most cases can be identified not despite of, but because of the lack of medical terms. The number of reports labeled seems to be sufficient for the task, but it may have been improved a bit if more reports had been labeled.

A rather ironic part of the labeling of invalid reports is that it did not use an active learning system. The analysis of the invalid reports was completed before the work with active learning started. While the active learning system dealt with in this report focused on multi-label data, it could have been beneficial to use. Evaluating the categorization of invalid reports by accuracy, precision, recall and F1-score are fairly standard. The metrics have been used in a lot of text classification and information retrieval research [1, 6], and should enable comparisons of the results with other sources.

The entire first and second experiments were based on the author's implementation of the algorithms described in Section 2.2. These were based on the algorithms in the papers, but may contain bugs or misinterpretations. Exposure to public scrutiny may have been able to find any faults, and in turn make the results more reliable, since any bugs most definitely effect the results of the study.

The analysis of distributions is evaluated using some non-standard techniques. Comparing imbalanced datasets in binary classification can easily be done with the ratio between the two classes. This is harder to do in multi-label classification. Another common approach is to measure how well a model performs, with F_1 -score for example. This was already done in research question 2, and does not measure the uniformity of the distribution explicitly, but rather implicitly by assuming that it will make the models perform better. The evaluations that this report uses are instead fairly non-standard, but focuses on being intuitive in measuring how uniform the set is. So the validity of the results here can rightly be criticized. By using non-standard metrics, an argument can be made that it does not accurately achieve what it aims to do. On the other hand, they are clearly defined, and make intuitive sense. If the most common categories make up most of the labels assigned, the dataset is probably not very well balanced. The ratio between the smallest and the biggest category is an attempt to generalize the imbalance ratio used when measuring the binary case. It too makes intuitive sense, a big ratio between them indicates that there is at least a couple of classes where the labeled set is imbalanced.

The sources used in the thesis are a mix of scientific articles and books. One theme amongst the active learning sources is that some of them are quite old. Some papers, such as the one by Tong et al. [38], is from the early 2000 but provided a lot of the foundation that new techniques are based on. Research relating to multi-label active learning is also relatively sparse, compared to multi-class or binary. Newer techniques often seem to be focused on specific enhancement for using them with images. Besides that, sources used to provide an overview of the field, such as Settles [34] and Tong et al. [37] are rather well cited, being cited by a couple of thousand papers each.

5.3 Related Work

Using topic modeling for various clinical applications has been done before. Topic modeling has been a popular approach for this purpose since clinical data often is in the form of free text. The resulting topics can also be interpreted by humans, which allow doctors to get more insight into the system. Sarioglu et al. [32] to represent clinical reports with topic vectors in order to classify them. Chan et al. [10] used topic models to analyze patient records and clinical reports from cancer patients. In their paper, they found relationships between the content of the notes on the patients, with the data that was available on the patients' genetic mutations. The interpretability of topics generated from an LDA model was studied on clinical reports by Arnold et al [3]. They evaluated how interpretable topics were based on how many topics the model used.

Active learning has been researched in text classification with different approaches. They can be seen as two categories: searching through the hypothesis space by using the uncertainty of a model, or by exploiting the structure of the data through clustering [12].

One of the common baselines for active learning is uncertainty sampling [22], that simply queries the label for the data point the model is most uncertain about. In [12] hierarchical clustering is used in an active learning system. The labels are queried from clusters where there is a lot of uncertainty when it comes to the majority label. By pruning the tree of clusters while querying for labels the goal is to obtain a pruning where each node mostly contains one label.

In [27] they also take advantage of a clustering to select the samples to be labeled in a two-class environment. They use that the data points closest to the centroids are the most important ones, and that most data points in one cluster have the same label. What this approach has in common with a lot of the current research is that it is treating single-label or binary classification problems, which cannot be directly applied to a multi-label scenario.

Research in [9] is dealing with the multi-label problem. That is the paper that developed the *binary version space minimization* strategy that is described in section 2.2.3. It simply takes the instance with the smallest margin among the binary classifiers, using the binary relevance scheme. The MMC strategy [42] that is described in section 2.2.4, and the adaptive active learning strategy [23] in section 2.2.5 are also techniques for managing the multi-label problem. MMC tries to find the greatest reduction for the estimated loss. While adaptive active learning combines an uncertainty measure with a measure of how the label cardinality differs. Singh et al. [36] is another multi-label active learning approach that simply takes the minimum average of the margin among the classifiers for a data point. For image classification, there has been some methods develop, for example [24, 28]. In [24] the goal is to, after making predictions, selecting the sample with the biggest mean loss. However experiments have shown that this is not as suitable for text classification [42]. In [28], the approach is to use pairs of labels and samples to present to the annotator, and the aim is to minimize the Bayesian classification error. Due to the fact that labeling for text classification is more time consuming than image classification, since you have to read an entire text, this approach is not suitable for text classification [42].

Active learning has been used to deal with the problem of imbalanced datasets before. In a binary classification setting, Ertekin et al. [13] used uncertainty sampling with SVMs to get a more balanced dataset. In [5], Attenberg et al. uses density based active learning to improve the class balance. However, it does not attempt to apply this in a multi-label setting.

5.4 The Work in a Wider Context

A discussion regarding how an active learning system affects the world around us is probably more interesting from a perspective of what it enables. By being able to obtain a high quality set of labeled reports with less effort, people can create powerful machine learning models with less time spent on gathering data. These systems can, and probably will, have a profound effect on the world. Besides the improvements it may make to healthcare overall, it can also effect the jobs of those working within the field. Replacing doctors with systems based on artificial intelligence may not happen any time soon. One key advantage humans have, with the current state of the field, is creativity and emotional intelligence. Both are things that are of great importance in healthcare and dealing with patients. However, the jobs of doctors may change drastically given the aid these systems can provide them with. They systems could, for example, help doctors in determining diagnoses. Jobs that are of a more administrative nature are in more danger of being replaced in the near future.

Another interesting aspect is the trust in these systems. In order for doctors to be able to trust the systems, the process by which the different models make decisions will have to be rather open. Besides the understandability of model, the storage and treatment of the data is another area where there have to be trust in the systems. Any computer system may be faced with breaches from an intruder, and the hospital records are often already digitized today. However, automating the systems may make it easier to enforce certain procedures for how the data should be treated. Something that might be of great importance going forward.



6 Conclusion

This thesis can be seen to have been done in two parts: filtering out the invalid reports, and identifying active learning strategies that could be used to improve the labeling process. Identifying and filtering out invalid reports was treated in the first research question. It resulted in a rather good separation of the two categories. Accomplishing this using only unsupervised techniques is possible, even if using them together with a supervised technique gave a better result. But manually identifying relevant topics and classifying based on this information was clearly a working approach. It did not achieve 100 % accuracy, so there is room for improvement, but the separation has to be seen as successful.

When it comes to the alternatives to labeling reports at random, the second research question, there are a few alternatives. The ones chosen for evaluation here were the ones that were adapted for the multi-label scenario. These were:

- Binary Version Space Minimization (BSVM)
- Maximum Loss Reduction with Maximum Confidence (MMC)
- Adaptive Active Learning (AAL)

Their performance can be summarized in that MMC performed worse in the early stages, while the adaptive approach worked the best. BSVM performed approximately the same as random sampling in the beginning. In the end, BSVM and MMC both achieved a higher accuracy and F_1 -score than AAL, at least after 2000 samples were added. All of the above mentioned strategies performed better than random sampling in the long run.

When it comes to the distribution of labels, the strategies all performed better than random sampling at all stages. After 2000 labels, BSVM and MMC had a more even distribution than AAL. While early on, MMC and the adaptive approach generated a lot more even label distribution than BSVM. So, in various degrees during the labeling process, the different strategies all made the dataset more uniform than labeling at random.

For future research, it would be interesting to look into how to further use the structure of the data in active learning strategies, besides only obtaining initial samples from clusters. An example of this would be to find a good way to adapt the approach described by Dasgupta et al. [12] to the multi-label case. With their approach to binary classification, it is rather easy to define a measure to see when a cluster overwhelmingly consisting of one class. However, if a data point can consist of any combination of labels it becomes a lot harder. Using the full

approach, including the usage of clusters to classify the points without another model, might be hard for this reason. Researching whether or not the hierarchical clustering could aid the selection of data points, instead of both selection and classifying, might be more approachable.

Another thing that would be of interest is to vary the text representation and classification models to see how they get effected by the different strategies. One example of this could be to use recurrent neural networks instead of SVM's, and see if the different active learning strategies affects the models differently. For the text representation, word2vec or latent topic vectors could be used instead of bag of words to highlight different features of the text. These methods described in this thesis should work well for other media such as images too, but if these were to be studied it would open up for some other techniques as well. Something that could be an interesting area for future research.



Bibliography

- [1] Charu C Aggarwal and ChengXiang Zhai. “A survey of text classification algorithms”. In: *Mining text data*. Springer, 2012, pp. 163–222.
- [2] Charu C Aggarwal and ChengXiang Zhai. “A survey of text clustering algorithms”. In: *Mining text data*. Springer, 2012, pp. 77–128.
- [3] Corey W Arnold, Andrea Oh, Shawn Chen, and William Speier. “Evaluating topic model interpretability from a primary care physician perspective”. In: *Computer methods and programs in biomedicine* 124 (2016), pp. 67–75.
- [4] David Arthur and Sergei Vassilvitskii. “k-means++: The advantages of careful seeding”. In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics. 2007, pp. 1027–1035.
- [5] Josh Attenberg and Seyda Ertekin. “Class imbalance and active learning”. In: *inde Imbalanced Learning: Foundations, Algorithms, and Applications* (2013), p. 101149.
- [6] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [7] David M Blei, Andrew Y Ng, and Michael I Jordan. “Latent dirichlet allocation”. In: *Journal of machine Learning research* 3. Jan (2003), pp. 993–1022.
- [8] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. “Learning multi-label scene classification”. In: *Pattern recognition* 37.9 (2004), pp. 1757–1771.
- [9] Klaus Brinker. “On active learning in multi-label classification”. In: *From Data and Information Analysis to Knowledge Engineering*. Springer, 2006, pp. 206–213.
- [10] Katherine Redfield Chan, Xinghua Lou, Theofanis Karaletsos, Christopher Crosbie, Stuart Gardos, David Artz, and Gunnar Ratsch. “An empirical analysis of topic modeling for mining cancer clinical notes”. In: *Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on*. IEEE. 2013, pp. 56–63.
- [11] Steven P Crain, Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. “Dimensionality reduction and topic modeling: From latent semantic indexing to latent dirichlet allocation and beyond”. In: *Mining text data*. Springer, 2012, pp. 129–161.
- [12] Sanjoy Dasgupta and Daniel Hsu. “Hierarchical sampling for active learning”. In: *Proceedings of the 25th international conference on Machine learning*. ACM. 2008, pp. 208–215.

- [13] Seyda Ertekin, Jian Huang, Leon Bottou, and Lee Giles. “Learning on the border: active learning in imbalanced data classification”. In: *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM. 2007, pp. 127–136.
- [14] Thomas L Griffiths and Mark Steyvers. “Finding scientific topics”. In: *Proceedings of the National academy of Sciences* 101.suppl 1 (2004), pp. 5228–5235.
- [15] Haibo He and Edwardo A Garcia. “Learning from imbalanced data”. In: *IEEE Transactions on knowledge and data engineering* 21.9 (2009), pp. 1263–1284.
- [16] Thomas Hofmann. “Probabilistic latent semantic analysis”. In: *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc. 1999, pp. 289–296.
- [17] Jing Jiang. “Information extraction from text”. In: *Mining text data*. Springer, 2012, pp. 11–41.
- [18] Thorsten Joachims. “Text categorization with support vector machines: Learning with many relevant features”. In: *European conference on machine learning*. Springer. 1998, pp. 137–142.
- [19] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. Vol. 344. John Wiley & Sons, 2009.
- [20] Rémi Lebrete and Ronan Collobert. “Word emdeddings through hellinger PCA”. In: *arXiv preprint arXiv:1312.5542* (2013).
- [21] Omer Levy and Yoav Goldberg. “Neural word embedding as implicit matrix factorization”. In: *Advances in neural information processing systems*. 2014, pp. 2177–2185.
- [22] David D Lewis and William A Gale. “A sequential algorithm for training text classifiers”. In: *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc. 1994, pp. 3–12.
- [23] Xin Li and Yuhong Guo. “Active Learning with Multi-Label SVM Classification”. In: *IJCAI*. 2013, pp. 1479–1485.
- [24] Xuchun Li, Lei Wang, and Eric Sung. “Multilabel SVM active learning for image classification”. In: *Image Processing, 2004. ICIP’04. 2004 International Conference on*. Vol. 4. IEEE. 2004, pp. 2207–2210.
- [25] Oscar Luaces, Jorge Díez, José Barranquero, Juan José del Coz, and Antonio Bahamonde. “Binary relevance efficacy for multilabel classification”. In: *Progress in Artificial Intelligence* 1.4 (2012), pp. 303–313.
- [26] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).
- [27] Hieu T Nguyen and Arnold Smeulders. “Active learning using pre-clustering”. In: *Proceedings of the twenty-first international conference on Machine learning*. ACM. 2004, p. 79.
- [28] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, and Hong-Jiang Zhang. “Two-dimensional active learning for image classification”. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–8.
- [29] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. “Classifier chains for multi-label classification”. In: *Machine learning* 85.3 (2011), p. 333.
- [30] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.
- [31] Erik F Tjong Kim Sang and Fien De Meulder. “Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition”. In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4* (2003).

- [32] Efsun Sarioglu, Kabir Yadav, and Hyeong-Ah Choi. “Topic modeling based classification of clinical reports”. In: *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*. 2013, pp. 67–73.
- [33] Hinrich Schütze and Craig Silverstein. “Projections for efficient document clustering”. In: *ACM SIGIR Forum*. Vol. 31. SI. ACM. 1997, pp. 74–81.
- [34] Burr Settles. “Active learning”. In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6.1 (2012), pp. 1–114.
- [35] Carson Sievert and Kenneth Shirley. “LDAvis: A method for visualizing and interpreting topics”. In: *Proceedings of the workshop on interactive language learning, visualization, and interfaces*. 2014, pp. 63–70.
- [36] Mohan Singh, Eoin Curran, and Pádraig Cunningham. “Active learning for multi-label image annotation”. In: *Proceedings of the 19th Irish Conference on Artificial Intelligence and Cognitive Science*. 2009, pp. 173–182.
- [37] Simon Tong. *Active learning: theory and applications*. Stanford University, 2001.
- [38] Simon Tong and Daphne Koller. “Support vector machine active learning with applications to text classification”. In: *Journal of machine learning research* 2.Nov (2001), pp. 45–66.
- [39] Grigorios Tsoumakas and Ioannis Katakis. “Multi-label classification: An overview”. In: *International Journal of Data Warehousing and Mining* 3.3 (2006).
- [40] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. “Mining multi-label data”. In: *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 667–685.
- [41] Vladimir Vapnik. *Statistical learning theory*. 1998. Wiley, New York, 1998.
- [42] Bishan Yang, Jian-Tao Sun, Tengjiao Wang, and Zheng Chen. “Effective multi-label active learning for text classification”. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2009, pp. 917–926.