Labelling Clinical Reports by with Active Learning and Topic Modeling

Uppmärkning av kliniska rapporter genom active learning och topic modeller

Simon Lindblad

Supervisor : Marco Kuhlmann
Examiner : Arne Jönsson

External supervisor : Mikael Nilsson

Abstract

Abstract.tex

Acknowledgments

`Acknowledgments.tex`

Contents

List of Figures

List of Tables

1 Introduction

The world's population is growing each year. Making healthcare more efficient and robust is of great importance in order to handle the challenges that arises with a growing population. 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 is growing it is of utmost importance to ensure that the quality of diagnoses remain high, and to minimize the risk of missing some critical piece of information. Taking advantage of the available medical information is key to create aforementioned systems.

Information pertaining to a patient's diagnosis is often in the form of written clinical reports. This is one example of information that can be utilized in automated systems, by extracting information from old reports, the process of writing new ones can be enriched. 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. Machine learning 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.

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 text classification, there exists a set of inputs, in this case text data, that already has a category assigned to it. This data is then used to fit the model so that it later make predictions for inputs that it has not yet been exposed to. One model that have been shown to be successful in text classification is Support Vector Machines (SVM) [17, 1, 34].

In order to fit a machine learning model to predict categories for clinical reports, we need a set of already categorized, or labeled, data. That is, we need to assign categories to an existing set of clinical reports. It is often the case that text data is widely available, but it is harder to come by data that is already labeled. Obtaining high quality data is important to use in machine learning systems, both in healthcare and other areas. Since the models require a sufficient amount of labeled reports, the task of labeling them can be cumbersome. Especially 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.

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 think will help the model improve the most. 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 models 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.

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

1.1 Motivation

This thesis is carried out at Sectra Medical Imaging IT Solutions AB, as a part of their research group. They are currently pursuing a research project with Region Skåne, which is responsible for the healthcare in Skåne, that is located in southern Sweden. The intention behind the project is to use machine learning techniques to, among other things, be able to suggest categories to doctors while they are writing medical reports. Another case is to use the categories of documents to present doctors with medical reports that are handling similar cases from the past. With this information the doctors could get an extra quality assurance check in their diagnostic flow.

In order to build these system, you need a substantial amount of labeled clinical reports. Therefore, the purpose of this thesis is to increase the quality and efficiency of the process of labeling these reports. This will be done by using unsupervised learning techniques such as topic models and clustering to first remove documents that are not supposed to be labeled. These invalid reports are documents that describe patients never showing up for a scan, deceased patients and patients being moved to a different hospital, among others. For the labeling, a system is to be built to use active learning in conjunction with the aforementioned unsupervised techniques to increase the quality of the labeled documents. In the work that they have done so far in the research project, the doctor that primarily worked with the labeling of reports stated that the distribution over the labeled categories is very skewed. The vast majority of labeled documents were assigned the same few categories. This in turn leads the models to require a lot of labeled samples to work well. By using active learning we can reduce the number of labeled samples needed to obtain an accurate model.

1.2 Aim

The purpose with this thesis project is to evaluate different solutions to increase the quality of labeled reports, and thereby reducing the amount of them needed for use within the project.
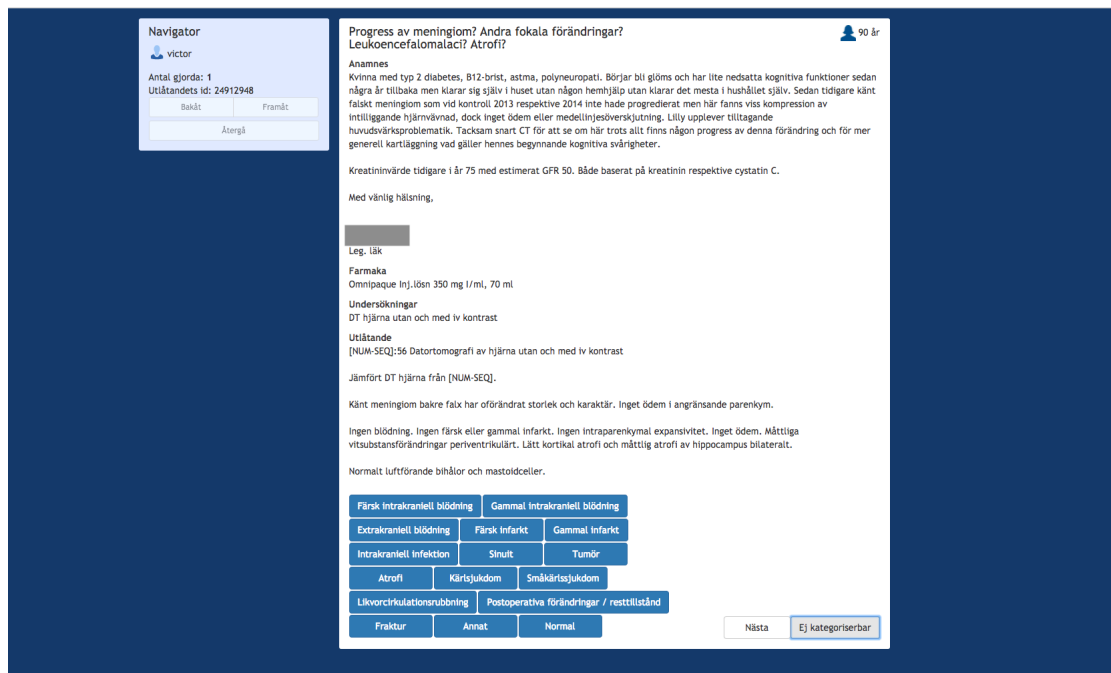
Figure 1.1: A screenshot of the interface used to label clinical reports at Sectra

Resulting from this will be a complete, standalone, system for labeling reports. The reports are interactively queried so a user can label the reports that are deemed most useful by the system. This will then be used together with an existing web interface that Sectra created for the purpose of labeling reports, which can be seen in Figure **??**.

1.3 Research questions

The specific research questions that this thesis will treat is presented here. They will be the main focus of study.

1. *Is it possible 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 another hospital. These reports does not contain any information of value from a medical point of view and should not be considered in the report labeling process.

   Unsupervised machine learning models such as topic modeling does by definition not require any labeled documents to train on. If it is possible to, without any such data, group these invalid reports together and remove them from the process before a doctor is presented with them that would be an additional hurdle removed from the process.

2. *What active learning strategies are good alternatives to sample documents at random in a multi-label document labeling system? How well does these alternatives perform?*
   How we are choosing the documents to be sampled is important. 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 the system requiring a lot of labeled documents to gain a high accuracy with reports of less common categories.

   If the decision boundaries of our models can be used to pick documents that would be more informative, the number of labeled documents could be reduced and still gain

the same accuracy. Another approach to selecting the documents to sample is to take advantage of the underlying structure of the data through clustering.

The algorithms to be evaluated will be based on the models certainty, as well as taking advantage of the underlying structure of the data. When choosing the algorithm to use, there are several different factors that will affect the final results and therefore needs to be taken into consideration. How well the models perform on the data is a rather obvious one – evaluating the models based on accuracy, precision, recall and f1-score. But they also need to be able to query documents in a reasonable time, if it is expensive to label reports it is likely to be expensive to wait for the reports to be queried. Choosing reports in batches and if the algorithm needs a big initial set of labeled reports are other factors that will be evaluated.

3. *How does the algorithms from question 2 effect 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 not very balanced. There are certain categories, like the one describing that everything is okay with the patient, that is a lot more common than other more rare illnesses. Even though the original data may be very imbalanced, selecting samples that contains a better balance between the different samples could improve the performance of the models.

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. The best balance being the one with the smallest ratio between the most common, and the least common labels.

1.4 Delimitations

Even though the sampling strategies are evaluated objectively on the Reuters dataset, the applicability of the techniques on clinical data is only evaluated by one physician, on the one dataset provided by Sectra.

1.5 Structure of the Report

The next chapter covers the background theory that is relevant for the thesis. After the theory, the methodology used is described, which is followed by a chapter covering the results. The method and results are then discussed in Chapter 5. Finally, Chapter 6 presents the conclusions.

2 Theory

In this section the theory behind the techniques used during the thesis work will be presented. The first part will go through techniques used to process the data and perform an exploratory analysis. After that text classifications, primarily with SVM, will be covered. The last section contains an overview of the field of active learning, as well as a comparison of some different active learning techniques for multi-labeled data.

2.1 Text Processing using Unsupervised Techniques

Techniques in machine learning that does not require you have a categorized or labeled set of data is called unsupervised. They use the structure of the data to obtain the information to use when processing it. 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.

When working with text, it needs to be represented in a way that allows the models to work with it effectively. *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. That is, a document is represented by the number of occurrences of the different words. The representation of a document therefore becomes very high-dimensional, there is one dimension for each word in the vocabulary. 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. Another drawback from this approach is that a word in a written language can be used to express several different thoughts, and one thought can be expressed using several different words. However, it is easy to work with, and is used when performing topic modeling among other things.

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. Considering the text "Pattern Recognition and Machine Learning" using a bigram (n-gram with n=2), would result in the tokens: "Pattern Recognition", "Recognition and", "and Machine", and "Machine Learning".

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 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 is Latent Dirichlet Allocation (LDA) [11]. Another topic model that preceded LDA is Probabilistic latent semantic analysis (PLSA) [15]. 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: $D = \{w_1, w_2, \ldots, w_M\}$.

- 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 topic i.

- V denotes the number of words in the vocabulary.

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.
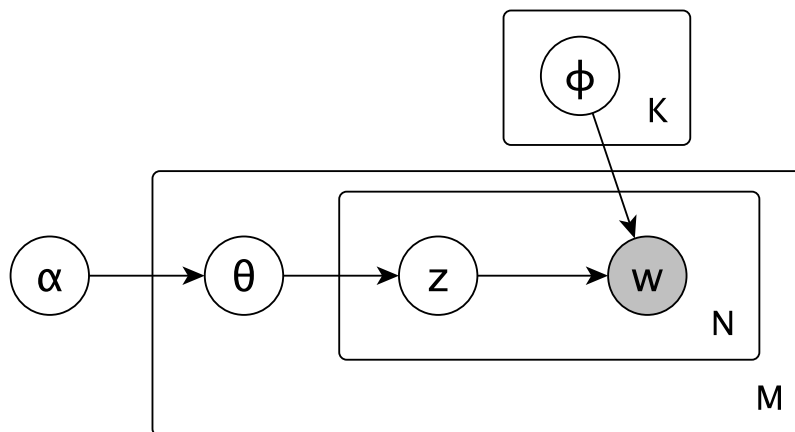
Figure 2.1: Diagram of the LDA model.

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.

$$\Phi_i \sim Dir(\beta) \tag{2.1}$$

$$p(\Phi_i|\beta) = \frac{\Gamma(V\beta)}{\Gamma(\beta)^V} \prod_{v=1}^{V} \phi_{iv}^{\beta-1} \tag{2.2}$$

Here, Γ is the gamma function.

- **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.

$$\Theta_d \sim Dir(\alpha) \tag{2.3}$$

$$p(\Theta_d|\alpha) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_{di}^{\alpha_i-1} \tag{2.4}$$

- 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.

$$z_{dn} \sim Multinomial(\Theta_d) \tag{2.5}$$

$$p(z_{dn} = i|\Theta_d) = \theta_{di} \tag{2.6}$$

  - **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 $\phi_{z_{dn}}$.

$$w_{dn} \sim Multinomial(\Phi_{z_{dn}}) \tag{2.7}$$

$$p(w_{dn} = v|z_{dn} = i, \Phi_i) = \phi_{iv} \tag{2.8}$$

The LDA model identifies topics from different terms that occur together. 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]. In collapsed Gibbs sampling, θ and ϕ are marginalized out. It works by repeatedly sampling the topic assignment z_{dn} for each token, conditioned on the assignments for the other tokens. Variational Bayes works by using simpler single-variable models to approximate the LDA. As a consequence, it disregards any dependencies between the variables. This is the approach used in the original LDA paper [7].

Collapsed Gibbs

To be written

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 [18, 2]. Since text data is 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 does not work as 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 [2], as well as Euclidean distance. Two different approaches to distance-based clustering are distance-based partitioning, and agglomerative hierarchical clustering.

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 can be seen as:

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, 29]. 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].

K-means is commonly used with Euclidean distance, which is defined on two n dimensional points $p1$ and $p2$ as:

$$d(p1, p2) = \sqrt{(p1_1 - p2_1)^2 + (p1_2 - p2_2)^2 + \cdots + (p1_n - p2_n)^2} \tag{2.9}$$

Word Embeddings with Word2Vec

To be written

2.2 Text Classification

Text classification is a widely studied field within Computer Science. It is an important problem in supervised machine learning, and it is the task of assigning one or more classes to a given text document [1]. The problem 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. 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) Initial seeds

(b) Iteration 1

(c) Centroids after iteration 1

(d) Iteration 2

(e) Centroids after iteration 2
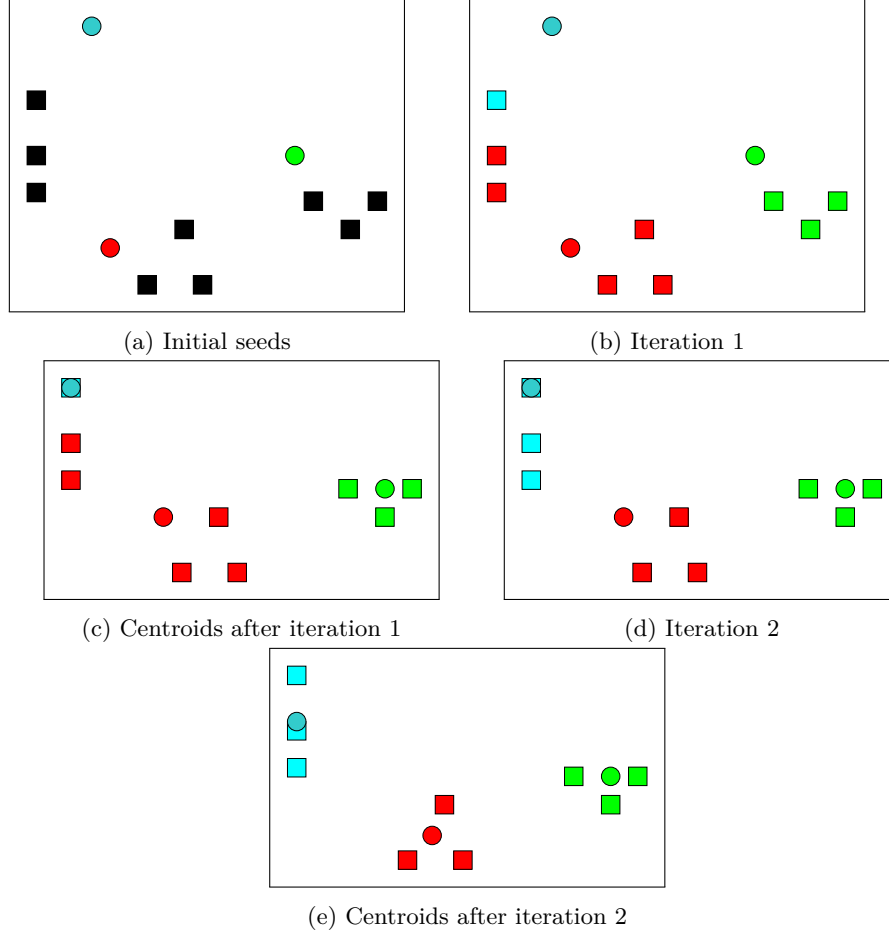
Figure 2.2: (a) to (e) shows iterations of K-means until convergence. In (e) it can be seen that the new centroids capture the same documents as the previous iteration, and we have converged.

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 [34]. Example of use cases for text classification is categorization of news articles, document retrieval and email filtering. There exists 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.

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:

$$y(x) = w^T \phi(x) + b \tag{2.10}$$

The sign of $y(x)$ determines the label assigned to x.

SVMs work by trying 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. The hyperplane that gives us the maximum margin can be found by:

$$\arg\min_{w,b} \frac{1}{2}\|w\|^2 \tag{2.11}$$

In order to allow for better generalization, and for data that isn't completely linearly separable, SVMs make use of slack variables ξ_n 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:

$$\arg\min_{w,b} \frac{1}{2}\|w\|^2 + C \sum_n \xi_n. \tag{2.12}$$

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

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 you have more than 2 labels, but each document is only assigned one [35].

A common way of solving multi-label classification problems is the Binary Relevance method [25, 8, 22]. It is a way of transforming the multi-label classification problem into several different binary ones. With Binary Relevance you fit one classification model per label in your data. Each of these classifiers are then predicting whether or not the document is associated with the corresponding label or not.

2.3 Active Learning

Conventional machine learning systems use a set of available data to find a hypothesis that can explain the patterns. The purpose of active learning is to allow a system to *select* the data that it wants labeled, and therefore the data it wants to be trained on [30]. An active learning system samples a document to be labeled from a pool of unlabeled data, and then queries an oracle (often a human annotator) to get the label for that document. By being able to decide what data to to label and use, the goal is that the system can achieve better results, and that the data will be of higher quality. To get a better understanding of how the process works, these are the steps commonly iterated over until enough samples have been labeled:

1. Evaluate the samples in the unlabeled pool based on a particular measure that the querying strategy defines.

2. The selected samples are presented to an oracle that is queried for the labels. This oracle is commonly 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 querying strategy in order to select samples to be label in step 2

A model of the active learning system can be seen in Figure 2.3.

In several different domains, data is readily available and easy to come by. But even if the data is abundant, labels for the data is often harder and more expensive to come by [30], especially when it comes to multi-label problems.

The next section will describe different ways to access the documents in active learning systems, followed by some theory of how the samples relate to the hypothesis space. After that some concrete methods for selecting the samples to queried are described and compared.
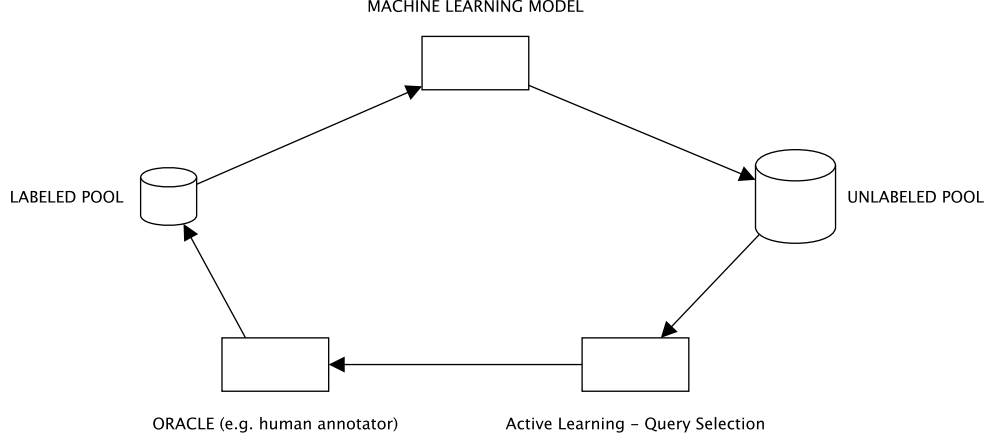
Figure 2.3: An overview of an active learning system

Pool-Based Sampling

The main focus 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 exists a large ready pool of data, where only a small portion is labeled [19, 30]. The samples to be labeled is then selecting by evaluating the entire pool of unlabeled data, and select the most appropriate one based on a defined utility measure. If the entire pool is large, a subset could be used instead. For applied active learning, pool-based sampling seems to be the most popular choice [20], 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 the individuality of the decisions in stream-based selective sampling, where you draw 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 you can consider the entire dataset. Pool-based is therefor the sampling technique that will be considered in this paper.

Searching the Hypothesis Space

In machine learning 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 the *hypothesis space*. Following the SVM example, the hypothesis space would be the set of SVM's with the different values that are possible for the parameters. The hypothesis space is defined as:

$$\mathcal{H} = \left\{ f | f(x) = \frac{w * \phi(x)}{|w|} \right\} \tag{2.13}$$

where $w \in \mathcal{W}$ and $\mathcal{W}$ is our parameter space

The version space the subset of the hypothesis space The subset of the hypothesis space that in the feature space separates the data is called the version space, which is defined as:

$$\mathcal{V} = \left\{ f \in \mathcal{H} | y_i f(x_i) > 0 \forall i \in \{i \ldots n\} \right\} \tag{2.14}$$

So the version space therefore represents the different hypothesis 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 make it half the size in each iteration.

There exists a useful relationship between the feature space $\mathcal{F}$ and the hypothesis space $\mathcal{H}$ called the *version space duality* [34, 37]. It states that hyperplanes in the hypothesis 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 is called that has shown to be successful is *Uncertainty Sampling* [30]. 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. We want to select the points in the feature space that will reduce the amount of the valid hypotheses the most. Since SVMs 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 encompassed constraints set by the labeled points. What Uncertainty Sampling is predicting the values for the unlabeled points, and then choose the one that it is most uncertain about, the one closest to the decision boundary, to be labeled. Based on the version space duality, it is a good approximation for dividing the version space in two.

Binary Version Space Minimization

Binary Version Space Minimization is a generalization of uncertainty sampling, to make it work with multi-label data. The approach taken is to decompose the multi-label problem to several binary one-vs-rest tasks, like discussed in 2.2. 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.

Maximum Loss Reduction with Maximum Confidence

Maximum Los Reduction with Maximum Confidence(MMC) was developed by Yang et al [38]. The goal of the 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 [38]:

$$\widehat{\sigma_{D_L}} = \int_x \left(\sum_{x \in Y} L(f_{D_L}) P(y|x) \right) P(x) dx \tag{2.15}$$

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

$$\widehat{\sigma_{D_L}} = \frac{1}{|D_U|} \sum_{x \in D_U} \sum_{y \in Y} L(f_{D_L}) P(y|x) \tag{2.16}$$

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 [38]:

$$D_S^* = \arg\max_{D_S}(\widehat{\sigma_{D_L}} - \widehat{\sigma_{D'_L}}) = \arg\max_{D_S} \left(\sum_{x \in D_S} \sum_{y \in Y} (L(f_{D_L}) - L(f_{D'_L})) P(y|x) \right) \tag{2.17}$$

In their paper, Yang et al. 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 a 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. Since we can obtain the probability from our classification model for each label, we can sort them in descending order. The first m labels are then likely to have a high probability, while the rest a rather low probability.

Yang et al. [38] describes 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, train a logistic regression classifier with the sorted and normalized probabilities as features, and the number of labels as target.

4. With the trained 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.

The only task that's left is now to estimate the loss for the multi-label classifier. 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:

$$L(f) = \sum_{i=1}^{k} L(f^i) \tag{2.18}$$

where the loss of a single binary classifier is denoted as $L(f^i)$. With this definition, it only 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 [34, 38]. The version space's size can be computed with Equation 2.14. However, computing this

for each possible label set is expensive. By using the heuristic from [33], an approximation of the version space with the added label can be obtained from the current SVM classifiers' margin. The reduction rate after adding a the data point (x, y^i), where $y^i \in -1, 1$, can be expressed as follows [38, 33]:

$$\frac{L(f^i_{D_L+(x,y^i)})}{L(f^i_{D_L})} \approx \frac{V^i_{D_L+(x,y^i)}}{V^i_{D_L}} \approx \frac{1+y^i f^i_{D_L}(x)}{2} \tag{2.19}$$

$L(f^i_{D_L})$ does not involve the sample selected for labeling, so by writing the loss reduction as:

$$L(f_{D_L}) - L(f_{D'_L}) = \sum_{i=1}^{k} \left(L(f^i_{D_L}) - L(F^i_{D'_L}) \right)$$
$$= \sum_{i=1}^{k} \left(L(f^i_{D_L}) \dot{(} 1 - \frac{L(F^i_{D'_L})}{L(F^i_{D_L})}) \right) \tag{2.20}$$

it can be seen that focusing on the reduction rate is sufficient. By incorporating the result from Equation 2.19, the following approximation for the reduction rate is obtained:

$$\sum_{i=1}^{k} \left(\frac{1 - y^i f^i_{D_L}(x)}{2} \right) \tag{2.21}$$

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 [38]:

$$D^*_S = \arg\max_{D_S} \left(\sum_{x \in D_S} \sum_{i=1}^{k} \left(\frac{1 - \hat{y}^i f^i_{D_L}(x)}{2} \right) \right) \tag{2.22}$$

Adaptive Active Learning

In their paper, Li et al. [20] presents 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.

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 [20]. That is, separating the labels gets assigned to an instance and the ones that does not. In order to model the uncertainty of the prediction 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:

$$sep\_margin(x) = \min_{i \in \hat{y}^+} f'_{D_L}(x) - \min_{i \in \hat{y}^-} f'_{D_L}(x)$$
$$= \min_{i \in \hat{y}^+} |f'_{D_L}(x)| + \min_{i \in \hat{y}^-} |f'_{D_L}(x)| \tag{2.23}$$

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:

$$u(X) = \frac{1}{sep\_margin(X)} \tag{2.24}$$

Label Cardinality Inconsistency

Label Cardinality Inconsistency is based on 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 [35]. 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:

$$c(x) = \left\| \sum_{i \in \hat{y}^+} 1 - \frac{1}{N_L} \sum_{y \in Y_L} \sum_{i \in y^+} 1 \right\|_2 \tag{2.25}$$

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:

$$q(x, \beta) = u(x)^\beta \cdot c(x)^{1-\beta} \tag{2.26}$$

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 $\{0, 0.1, 0.2, \ldots, 1\}$. Then selecting β based on the most informative sample amongst the . Equation 2.26 shows the *approximate generalization error*, which is used to select the sample.

$$\epsilon(x) = \sum_{x \in D_U} \max_{i \in f(x)^+} \left(1 - f^i(x)\right) + \max_{i \in f(x)^-} \left(1 + f^i(x)\right) \tag{2.27}$$

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

$$x^* = \arg \min_{x \in D_U} \epsilon(x) \tag{2.28}$$

2.4 Evaluation Metrics

For classification or information retrieval systems the typical evaluation metrics to use are *precision*, *recall* and *recall* [16]. 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*.

	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

Precision is the percentage of the results found by the system that are correct [27]. Recall is the percentage of correct results in the dataset that are found by the system. Precision and recall are defined as follows by Van Rijsbergen [26]:

$$Precision = \frac{tp}{tp + fp} \tag{2.29}$$

$$Recall = \frac{tp}{tp + fn} \tag{2.30}$$

F-score is the harmonic mean between recall and precision, and is defined as [27]:

$$F = \frac{2 * precision * recall}{precision + recall} \tag{2.31}$$

Another metric that is used for evaluation is *accuracy*, which is defined as follows:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{2.32}$$

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 [36]. Then you use the sums to obtain the final score. Using the notation established above, the micro average is defined as [36]:

$$B_{micro} = B(\sum_{\lambda=1}^{k} tp_\lambda, \sum_{\lambda=1}^{k} fp_\lambda, \sum_{\lambda=1}^{k} tn_\lambda, \sum_{\lambda=1}^{k} fn_\lambda) \tag{2.33}$$

Macro-average on the other hand works by first calculating the binary measure, and then taking the average of all of them [36]. It is defined as:

$$B_{macro} = \frac{1}{k} \sum_{\lambda=1}^{k} B(tp_\lambda, fp_\lambda, tn_\lambda, fn_\lambda) \tag{2.34}$$

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* [36].

TODO: Write about perplexity

2.5 Related Work

Using topic modeling for various clinical applications has been done before. Topic modeling have been a popular approach for this purpose since clinical data often is in the form of free text. The resulting topics can be interpreted by humans, which allow doctors to get more insight into the system. Sarioglu et al. [28] 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 [19]. 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 [23] they also take advantage of a clustering to select the samples to be labeled in a two-class environment. They take advantage of 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 2.3. It simply takes the instance with the smallest margin among the binary classifiers, using It selects the data point that has the smallest margin among the binary classifiers, using the binary relevance scheme. The MMC strategy [38] that is described in section 2.3, and the adaptive active learning strategy [20] in section 2.3 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. [32] 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 [21, 24]. In [21] 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 [38]. In [24], 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 [38].

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.

3 Method

The task of making a better system for labeling clinical reports was approached with several text mining techniques, support vector machines and a few active learning querying strategies. At first, the framework and tools used in the system are described, followed by a description of the provided dataset. Finally, the experiments used to answer the research questions are presented.

3.1 Frameworks and Tools

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, nltk, scikit-learn* and *gensim*. All the plotting was done using the *seaborn* and *bokeh* libraries. *pyLDAvis* was used for some additional visualization purposes with regards to topic models.

However, when it comes to the 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 adaptions such as the number of items queried at each iteration, an active learning framework was created. The ground for this framework were the algorithms presented in Section 2.3.

3.2 Datasets

In this thesis, two different datasets were used. The dataset provided by Sectra, as well as Reuters-21578. The latter was used to be able to simulate a multi-label labeling process, to evaluate how well the different strategies work before being integrated into Sectra's system. Since the vast majority of the dataset from Sectra was unlabeled, this could not effectively been done using only that.

The set of reports provided by Sectra contained 1068904 different entries, where 493 were initially labeled. The entries were spread out over several files, stored in the JSON format. However, those labels were subject to change, so they were mainly used to see if there were

```
{
    'ExamId': 24550003,
    'ReportText': '',
    'Anamnesis': '',
    'Question': '',
    'PatientAlert': ' ',
    'ExamComment': None,
    'Canceled': False,
    'ExamName': '',
    'ExamCode': '81100',
    'PatientSex': 'MALE',
    'PatientAge': 71,
    'Urgent': -1,
    'Pharma': None
}
```

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

any 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 report written by the physician after the examination

3. **Anamnesis**: The patient's account of their medical history

4. **PatientAlert**: FILL IN

5. **ExamComment**: FILL IN

6. **Cancelled**: Whether or not the examination was Cancelled.

7. **ExamName**: FILL IN

8. **ExamCode**: FILL IN

9. **PatientSex**: FILL IN

10. **PatientAge**: FILL IN

11. **Urgent**: FILL IN

12. **Pharma**: FILL IN

The work was mainly concerned with the ReportText field, since it contains the response to the result of the examination. But the for the complete Active Learning system the Anamnesis was used as well. The labels that were initially assigned to these reports were: "Blödning", "Infektion", "Metabol", "Tumör", "Cysta", "Missbildning", "Syndrom", "Demens", "Hydrocefalus", "Infarkt", "Kärlsjukdom", "Trauma", "Systemsjukdom", "Inklämmning" and "Normal".

The distribution of labels among these initially labeled reports can be seen in Figure 3.2 Note that this is only a count of the individual labels, and the multi-label nature of the labeling is not taken into account in the histogram.

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. All experiments used the *ModApte* split of the dataset, which is commonly used and readily available. It splits the dataset into a defined
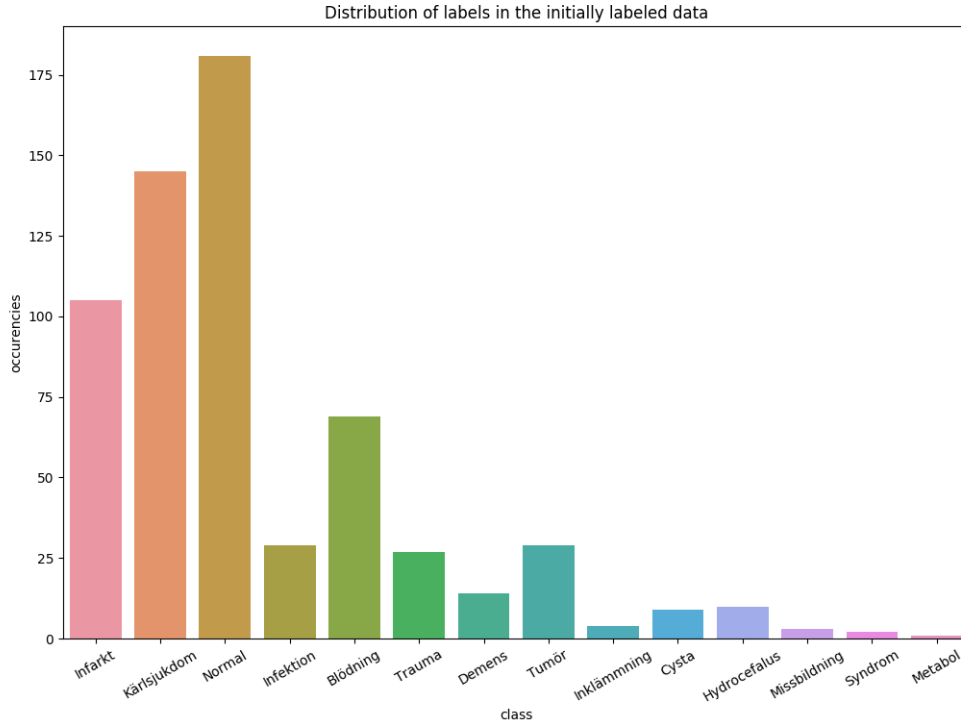
Figure 3.2: The distribution over the labels in the initial set of labeled data provided by Sectra

set of training and test documents, containing 7.769 and 3.019 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 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.

3.3 Pre-Processing and Text Representation

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

1. First, the fields of interest were extracted. In this case that was only the "ReportText" field.

2. White space and punctuation was stripped from the data.

3. All words were transformed into lowercase.

4. Filtered out the most common words, as well as very infrequent words. Specifically, words occurring less than in 1% of the documents, as well as 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 modelling before [34, 7, 9, 28].
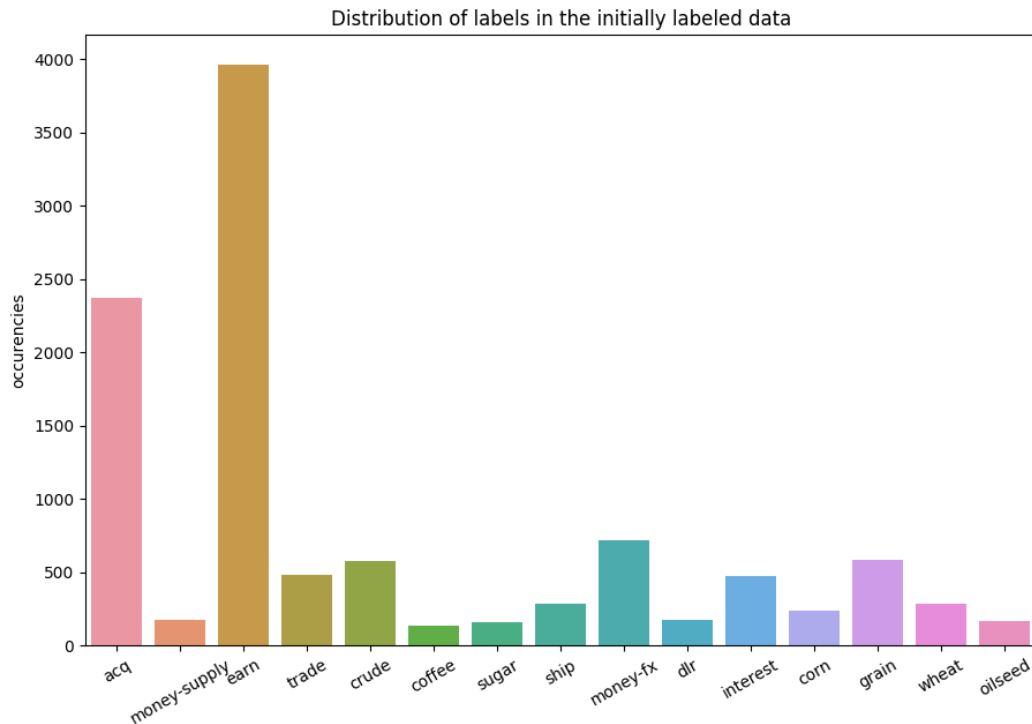
Figure 3.3: The distribution over the labels in the Reuters data

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 seeing the words that frequently occurred this list was extended to incorporate dataset-specific information. 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.

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

Most of these steps were performed in other reports dealing with text analysis in the form of classification or active learning [34, 7, 9, 28].

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 for them to work on. The representation chosen is a matrix of tokens count, so 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, besides the standalone tokens processed as described above. The additional tokens are bigrams. Bigrams are pairs of tokens (i.e. processed words), so the frequency of how often such a pair occurs in the document is stored alongside the regular one word tokens.

3.4 Exploratory Study

For the exploratory study we used the representation described in Section 3.3, but without the bigrams. The main goal of this phase was to get to know and to better understand the dataset. A part of this goal was to go through the fields for the different reports to see how
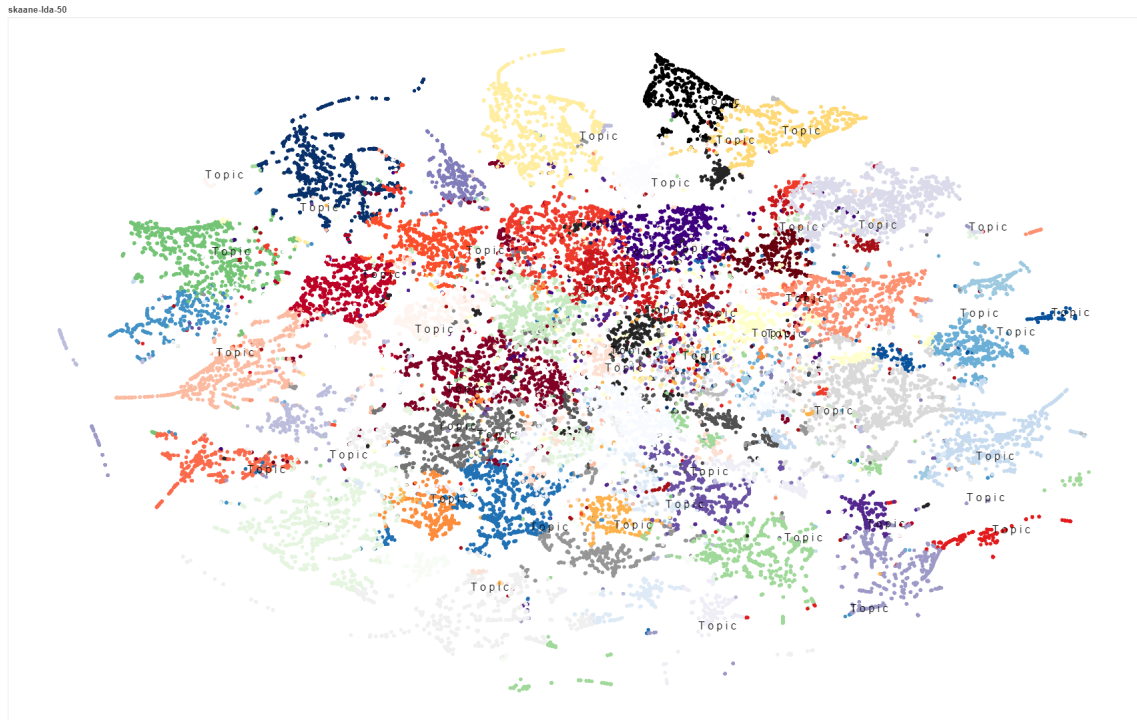
Figure 3.4: A 2D plot of the text data, where each point is colored by the most prominent topic

they worked and what values could be expected. Another big part of this was to visualize the dataset in different ways. 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, trying to retain as much variance as possible.

The first step was to fit an LDA model to the data. For the purpose of exploring the data, the number of topics were chosen to be a 100, with the hope of it not resulting in too granular topics that would be hard to manually analyze. In order to being able to visualize the data in a meaningful way, a subset of X reports were used to begin with. The data points were plotted in a 2D plot after reducing the dimensions using t-SNE. Since each data point is associated with several different topics, there exists several way of coloring each data point in order to gain an understanding of them. In this case, simply 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 3.4. Although it might be hard to interpret as a 2D plot in this report, using the *bookeh* library an interactive plot was generated, so hovering over each data point would show the content of the report, making this a convenient way to explore the data and the generated topics.

Samples of the generated topics can be seen in [FIGURE]. Another way to visualize the topics for inspection is using the techniques described by Sievert et al. [31]. 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 *pyLDAvis* can be seen in Figure 3.5.

A word2vec model was used on the entire dataset to evaluate see the relationship between terms and find possible synonyms. In order to find synonyms, all words in the dataset that had a similarity over 95% were manually inspected. This model was in addition to this used to identify names and other identifiers in the reports, as they would come up as similar entities from the model. Accomplishing this was done by exploring the data through an interactive plot, after using t-SNE to reduce the number of dimensions.
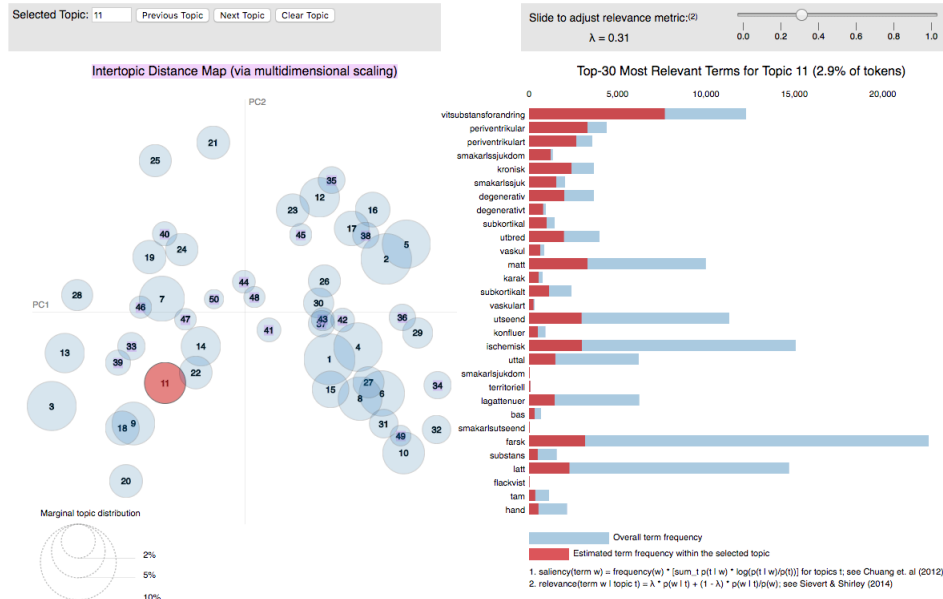
Figure 3.5: A way to visualize and analyze topics based on their relevance and frequency

3.5 Experiments to Answer the Research Questions

In this section the experiments used to answer the research questions are described. There is one experiment designed for each question. The first experiment is trying to identify reports that are deemed to be invalid, the second one is a study to see what are the alternatives to labeling data points at random, as well as a comparison of how well they perform according to certain metrics. Finally, the third one aims to compare how the distribution of labels in the labeled dataset compare between the different methods.

Filter Out Invalid Clinical Reports Using Topic Models and Clustering

Here, the task was to evaluate how well topic modeling and clustering could be used to filter out invalid reports. Invalid reports are here 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 simply did not want to go through with the examination for any reason.

Different topic model and cluster sizes were tested for this purpose. The topic model used was Latent Dirichlet Allocation, and the clustering algorithm used was K-means. The K-means clustering used the latent topic vectors produced by the topic model as representation when finding the clusters. As described in Section 2.1, in order to use the LDA model the number of topics has to be selected. The same applies to K-means, which is described in Section 2.1. Selecting the best model was done by evaluating different number of topics k_{LDA} and the number of clusters k_c. The topic models were evaluated with k_{LDA} set to 25, 50, 75, 100, 150, 200. Evaluating the clusters were done by combining the different topic models with the different cluster sizes to find the best combination. The different combinations can be seen in Table 3.1.

In order to evaluate how well they 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 allowed it to be marked as valid or invalid. At first X reports were labeled. After this set, the labels were skewed, containing only Y invalid reports, which makes up Y/X% of

ID	k_{LDA}	k_c	ID	k_{LDA}	k_c
1	25	25	19	100	25
2	25	50	20	100	50
3	25	75	21	100	75
4	25	100	22	100	100
5	25	150	23	100	150
6	25	200	24	100	200
7	50	25	25	150	25
8	50	50	26	150	50
9	50	75	27	150	75
10	50	100	28	150	100
11	50	150	29	150	150
12	50	200	30	150	200
13	75	25	31	200	25
14	75	50	32	200	50
15	75	75	33	200	75
16	75	100	34	200	100
17	75	150	35	200	150
18	75	200	36	200	200

Table 3.1: The different combination of topic model/k-means clusters that were evaluated.

the labels. In order to get a more balanced dataset, X_2 of the valid reports were dismissed at random.

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. 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. To determine which topic model and clustering technique that should be used in the filtering of invalid reports, their perplexity was compared and the models with the best perplexity was chosen. Perplexity is used in the original LDA paper by Blei et al. [7] to compare different number of topics.

In order to make the models able to separate the invalid reports from the valid reports, they had to be manually analyzed, since they were not fitted for this specific purpose (they are both unsupervised techniques). Approaching this in a way that would result in the models overfitted to the analyzed data could be hard since they are manually analyzed. To get an evaluation that was not as biased, the labeled reports were split into a training set to be analyzed and a validation set, containing 80% and 20% of the reports, respectively.

First, the LDA model was analyzed. This was done by inspecting the topics in the same way that was done in the exploratory study, Section 3.4. Based on the distribution of the most likely topics for the invalid reports in the traning set, topics with a high indication of a report being invalid was selected for further analysis. A combination these topics together with the length of the report text, as well as the number of topics assigned with a high probability to a report were used to determine whether or not a report was invalid or not.

Filtering out these reports with K-means results in a simpler method. K-means does not give any probability for its clusters, so filtering by the clusters must be done as a binary decision.

Alternatives to Label Reports at Random

This was done by first doing a literature study and exploring the relation between the initial set of labeled data with the structure of the data through clustering and topic analysis. At first, the labeled data was transformed using the LDA and k-means models. After that, they

were plotting in the same 2D space as before. The color of the labeled data was set based on the first label, after an instance's labels had been sorted.

Just as before this was an interactive plot, hovering over the data points revealed the report as well as the topics and the cluster assigned to the data point. The goal of this was to see if there existed a relationship between the topics/clusters and the labeled assigned to the data point. Even if the labels is not the same in the final system, knowledge of an existing relationship might still be exploited. Based on multi-label nature of the data and the results of this plot, active learning approaches were researched, with the goal of identifying methods that would be applicable in a multi-label setting. The research touched upon both methods that exploit the structure of the data, and methods that are purely uncertainty based.

After establishing the techniques that had some indication on providing a better labeling process than sampling documents at random, they were evaluated. In the end, the algorithms described in Section 2.3 were the ones selected for evaluation.

In order to provide a thorough evaluation of how well they techniques perform a set of already labeled reports were needed. For this reason, the Reuters-21578 dataset was used. The properties of the dataset, as well as a comparison between it and the clinical data provided by Sectra can be found in Section 3.2. The dataset is common in active learning research and has been used by Brinker et al. [9] and Yang et al. [38], among others. With this set of of labeled reports, a simulation could be used to compare the different strategies with different metrics. The metrics used were:

1. Accuracy

2. Micro recall

3. Macro recall

4. Micro precision

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. [38], Dasgupta et al. [12] and Li et al. [20], among others.

In addition to these metrics, the time it took to query samples with each model was also compared. This is because the doctors wanted a more uniform distribution over labels on the labeled samples, i.e. a smaller ratio between the most common label and the least common one.

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 goal, but instead offering a more comprehensive comparison.

The strategies compared were:

- *Binary Version Space Minimization*: Described in Section 2.3

- *Maximum Loss Reduction with Maximum Confidence*: Described in Section 2.3

- *Adaptive Active Learning*: Described in Section 2.3

ID	Active Learning Strategy	Initial Sampling	Initial Sample Size
1	Binary Version Space Minimization	Random	25
2	Binary Version Space Minimization	Random	50
3	Binary Version Space Minimization	Random	100
4	Binary Version Space Minimization	Random	200
5	Binary Version Space Minimization	Sampled from clusters	25
6	Binary Version Space Minimization	Sampled from clusters	50
7	Binary Version Space Minimization	Sampled from clusters	100
8	Binary Version Space Minimization	Sampled from clusters	200
9	Maximum Loss Reduction with Maximum Confidence	Random	25
10	Maximum Loss Reduction with Maximum Confidence	Random	50
11	Maximum Loss Reduction with Maximum Confidence	Random	100
12	Maximum Loss Reduction with Maximum Confidence	Random	200
13	Maximum Loss Reduction with Maximum Confidence	Sampled from clusters	25
14	Maximum Loss Reduction with Maximum Confidence	Sampled from clusters	50
15	Maximum Loss Reduction with Maximum Confidence	Sampled from clusters	100
16	Maximum Loss Reduction with Maximum Confidence	Sampled from clusters	200
17	Adaptive Active Learning	Random	25
18	Adaptive Active Learning	Random	50
19	Adaptive Active Learning	Random	100
20	Adaptive Active Learning	Random	200
21	Adaptive Active Learning	Sampled from clusters	25
22	Adaptive Active Learning	Sampled from clusters	50
23	Adaptive Active Learning	Sampled from clusters	100
24	Adaptive Active Learning	Sampled from clusters	200

Table 3.2: The different configurations of active learning strategies evaluated

When it comes to the initial samples that needs to be labeled in order for the strategies to base their selection on something. This sample were evaluated both by selecting it at random, and selecting then by sampling from the clusters. Sampling from the clusters was done by iterating over the clusters and selecting an equal number of data points from the clusters, at random. The clustering configuration was the one chosen in Section 3.5. The topic model selected in Section 3.5 was used. In addition, the number of clusters selected was 25, in order to get an equal number of reports from each cluster.

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. [38]. In their paper they tried quite large initial sample sizes. Here, the sizes evaluated are: 25, 50, 100, 200. 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. The different active learning configurations that were tried is displayed in Table 3.2 In total there were 18 configurations, based upon the three different methods.

3.6 Evaluating the Label Balance

The goal with the last experiment is to evaluate how the labels in the produced labeled dataset is 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.

The configurations used here are the same as in Table 3.2. Every iteration, the distribution of labels assigned are stored. An evaluation of how the distribution progresses can then be done by comparing the class imbalance between the different strategies at different stages.

4 Results

In this chapter the results are described. First, the outcome from the exploratory study is presented, followed by the different experiments. The first experiment, filtering out invalid reports, presents the evaluation of the topic model and k-means model used to filter out the reports, as well as the specific topics and clusters used in the process. In the second one, the methods considered and the decisions behind which ones that were appropriate are presented. Finally, the last section goes through the result of evaluating the different active learning techniques.

4.1 Exploratory Study

The goal with the exploratory study was to acquire a better understanding of the data, how it was structured and what kind of information might be extracted from it. [FIGURE FROM METHOD] displayed popular terms from a subset of the topics obtained from the topic model. Certain fields such as the cancelled 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 cancelled field was set to "false". After further manual analysis it was clear that the vast amount of invalid reports were contained within a few topics, something that is used in the first research question. The evaluation of more concrete relationships were done within the context of that experiment, and is presented in Section 4.2.

The word2vec model produced results that allowed for synonyms to be detected. By doing this, 420 pairs were discovered. The vast majority of these were names, medical terms that (some of which the author was unable to evaluate, and therefore excluded) as well as words that are used in similar contexts, which includes opposites like "left" and "right". Disregarding these, the synonyms and misspellings that were decided for use in the final system can be seen in Table 4.1. The original value was replaced with the new one in the final system.

In order to identify names from this word2vec model, it was plotted using the an interactive plot that allowed for exploring the data. Since names are commonly used in similar contexts, they would have similar attributes in the word embedding model. Figure 4.1 and Figure 4.2 show how this was done. 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.

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

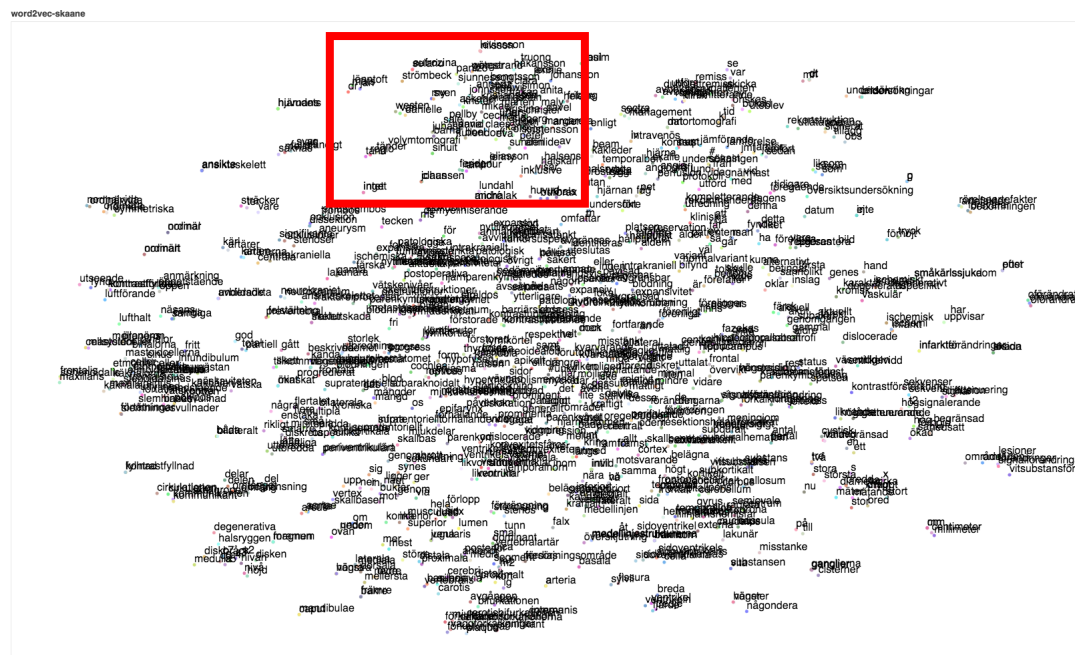Table 4.1: The synonyms, misspellings and shorts found in the data that the author could with assert with confidence.

Figure 4.1: A 2D plot of the full word2vec plot. Given the amount of terms used there is a lot to analyze.
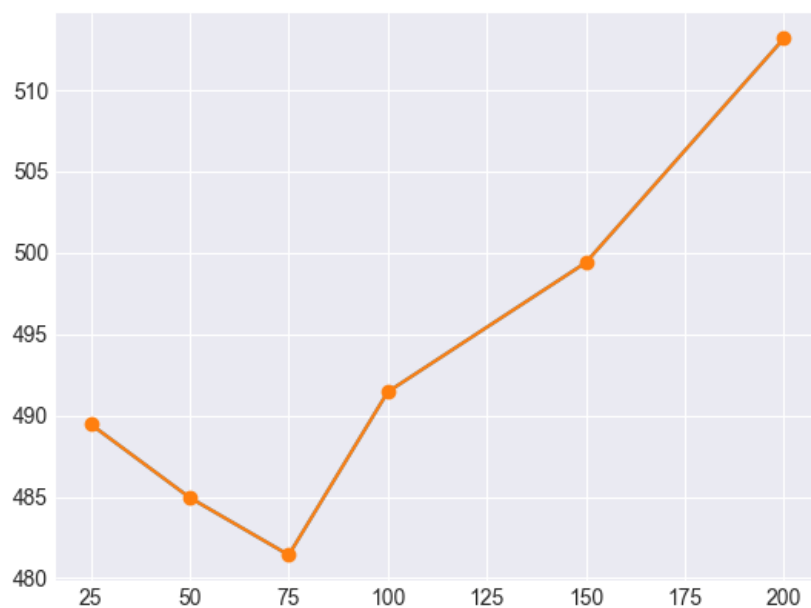
Figure 4.2: A 2D plot of the zoomed in word2vec plot. Most of the values here are names. This represents the red box zoomed in from Figure 4.1
.

4.2 Filter Out Invalid Clinical Reports Using Topic Models and Clustering

Still to be done: Add actual concrete results

The LDA models were evaluated by calculating the perplexity on the held-out set as described in Section 3.5. Perplexity for the evaluated models can be seen in Figure 4.3. Based on this, the selected model was the LDA model with 75 topics.

The next step was identifying the topics that were assigned to the invalid reports.

Each topic that is assigned to a report with a probability above 10% is considered to be a prominent topic. The distribution of the most likely topics for the invalid reports can be seen in Figure 4.4 and Figure 4.5. Figure 4.4 shows the distribution for the different topics where prominent topics for the invalid reports are counted. Figure 4.5 shows the distribution for the different topics where the topic with the highest probability for each invalid report is counted. A couple of topics, namely 2 and 28 stands out as the ones with by far the most invalid reports. But topics 4, 65 and 31 also show significant counts. In Figure 4.4 figure a few more topics, such as 19, shows a higher count.

The results of the analysis of the number of prominent topics assigned to each report can be seen in Figure 4.6 and Figure 4.7. Among the invalid reports, the most common number of prominent topic is 2, while it is 6 topics for the entire set of reports. For the entire set of reports, being assigned 2 topics is the 6th most common number. This result indicates that there is a relationship between the number of prominent topics assigned to each report, and wether that report is invalid or not.

Labels were determined to be invalid or not by the following criterion:

- Having either topic 2 or 28 among its most probable topic.

- Being shorter than 400 characters.

- Not having more than 5 prominent topics assigned to it.

The result of this was...

30

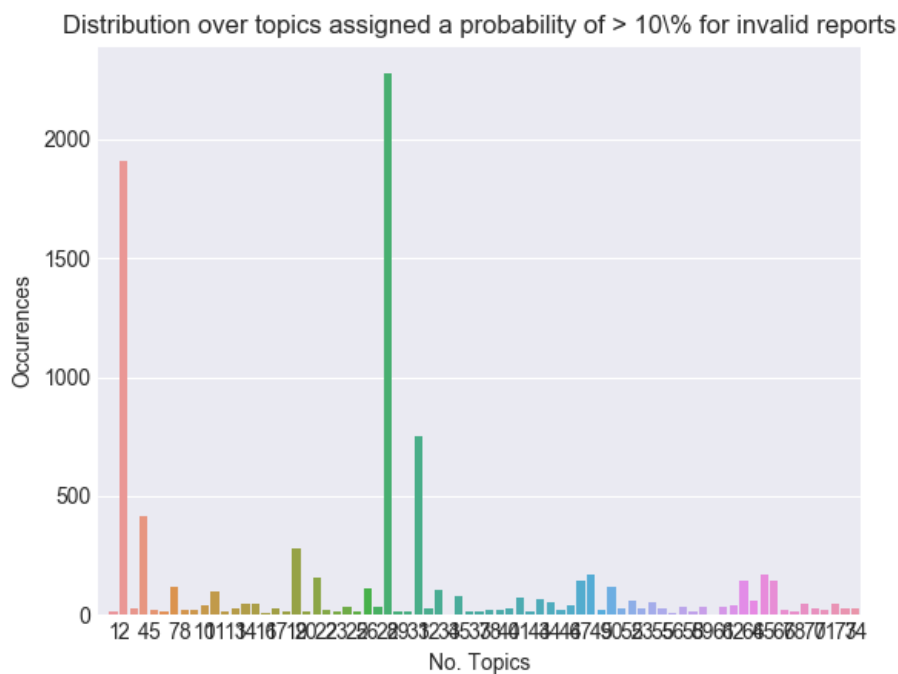Figure 4.3: The perplexity scores for the different LDA models

Figure 4.4: The distribution of invalid reports that got the topics assigned to them with a probability larger than 10%
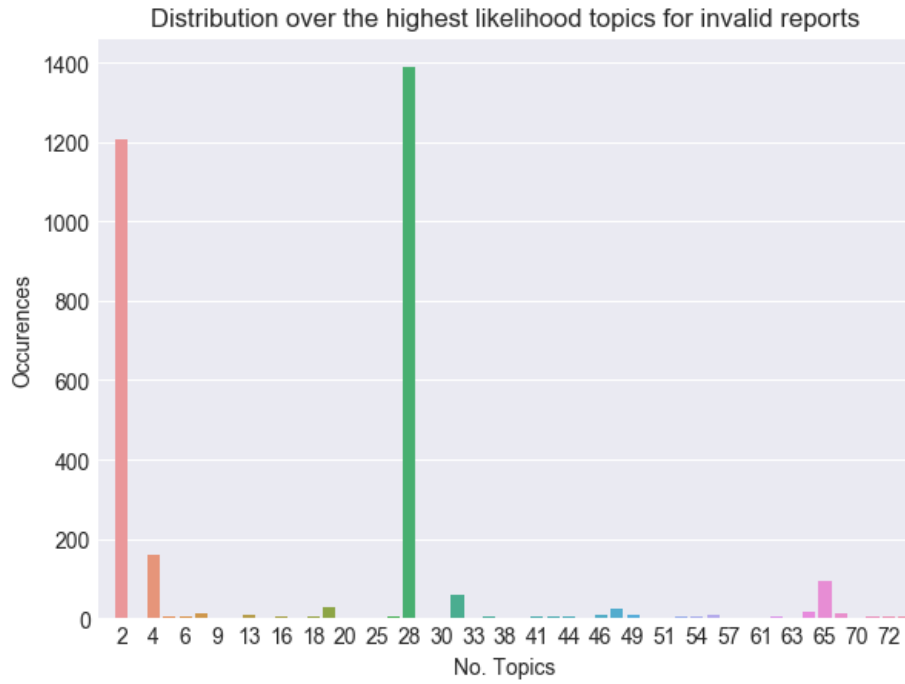
Figure 4.5: The distribution of invalid reports that got the topics assigned to them with the highest probability
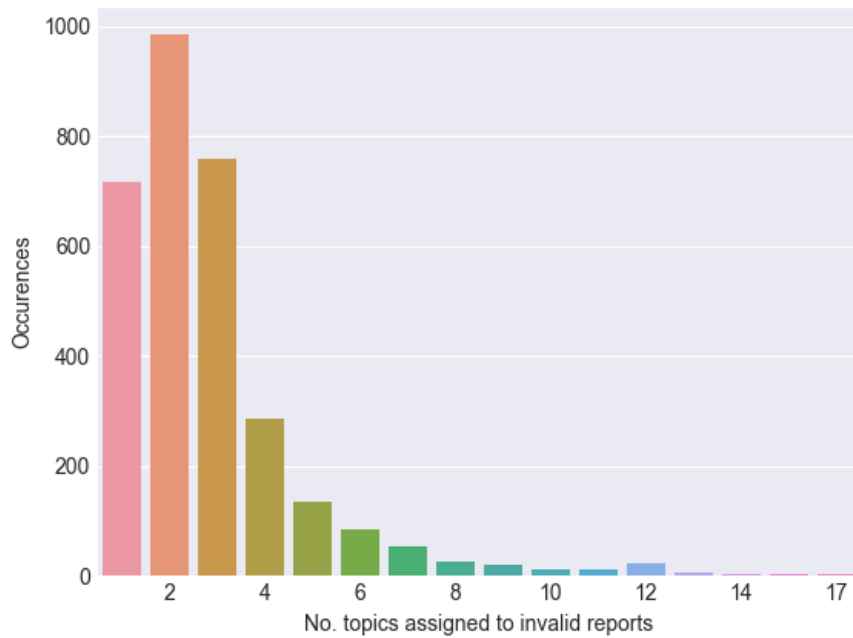

Figure 4.6: The distribution of the number of prominent topics assigned to the invalid reports
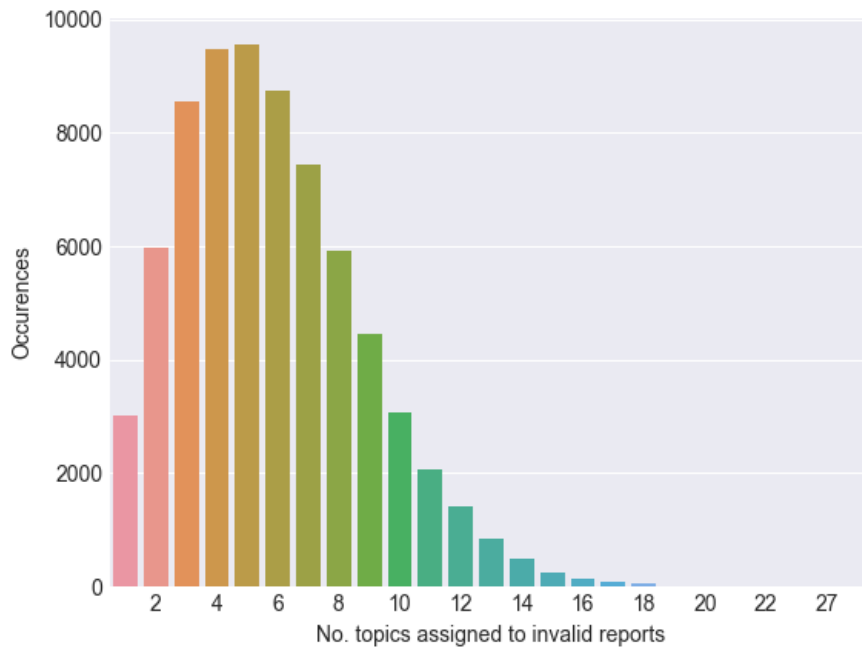
Figure 4.7: The distribution of the number of prominent topics assigned to the reports

4.3 Alternatives to Labeling at Random

Still to be done: Compare the different initial sample sizes (currently same plot)

The first technique used in this experiment was to explore and try to find a relationship between the initial set of labeled data and the inherit structure of it, obtained and visualized with the use of the LDA model. However, the relationship is not clear enough so that a cluster or topic could simply be mapped to a certain label. The resulting plot can be seen in Figure 4.8. From this plot it is clear that there is a grouping of labels. However, there exists plenty of overlap between different classes.

Based on the knowledge that there exists a pattern, the initial goal was to find some methods that could exploit this. Some active learning approaches using different forms of clustering, such as Dasgupta et al's approach using hierarchical clustering [12] would be good contenders. However, the method described by Dasgupta et al. is made for the single-label case with no obvious way of extending the technique into multi-label. The same applies to the density based technique suggested by Attenberg et al. [5].

Most of the active learning research seems to be focused on binary, or maybe multi-class classification. Therefor, the methods described in Section 2.3 where the ones decided on. Methods that are fully reliant on a models certainty, such as Binary Version Space Minimization [9] are tested. In addition to this, methods incorporating some information about the data in the form of label cardinality is included as well. These techniques are MMC and Adaptive Active Learning [38, 20]. An attempt to take advantage of the structure of the data is done by selecting the initial samples from different clusters, as described in Section 3.5.

The first metric for evaluation was Accuracy. A plot of this can be seen in Figure 4.9 to Figure **??**. It contains the accuracy evaluations for the different number of initial sample sizes. It can easily be seen that the adaptive learner performs significantly better for the first 400 added samples. However, after that MMC and Binary Version Space Minimization catches up to it, and surpasses it in terms of accuracy after around 800 labels. Both of them sustaining
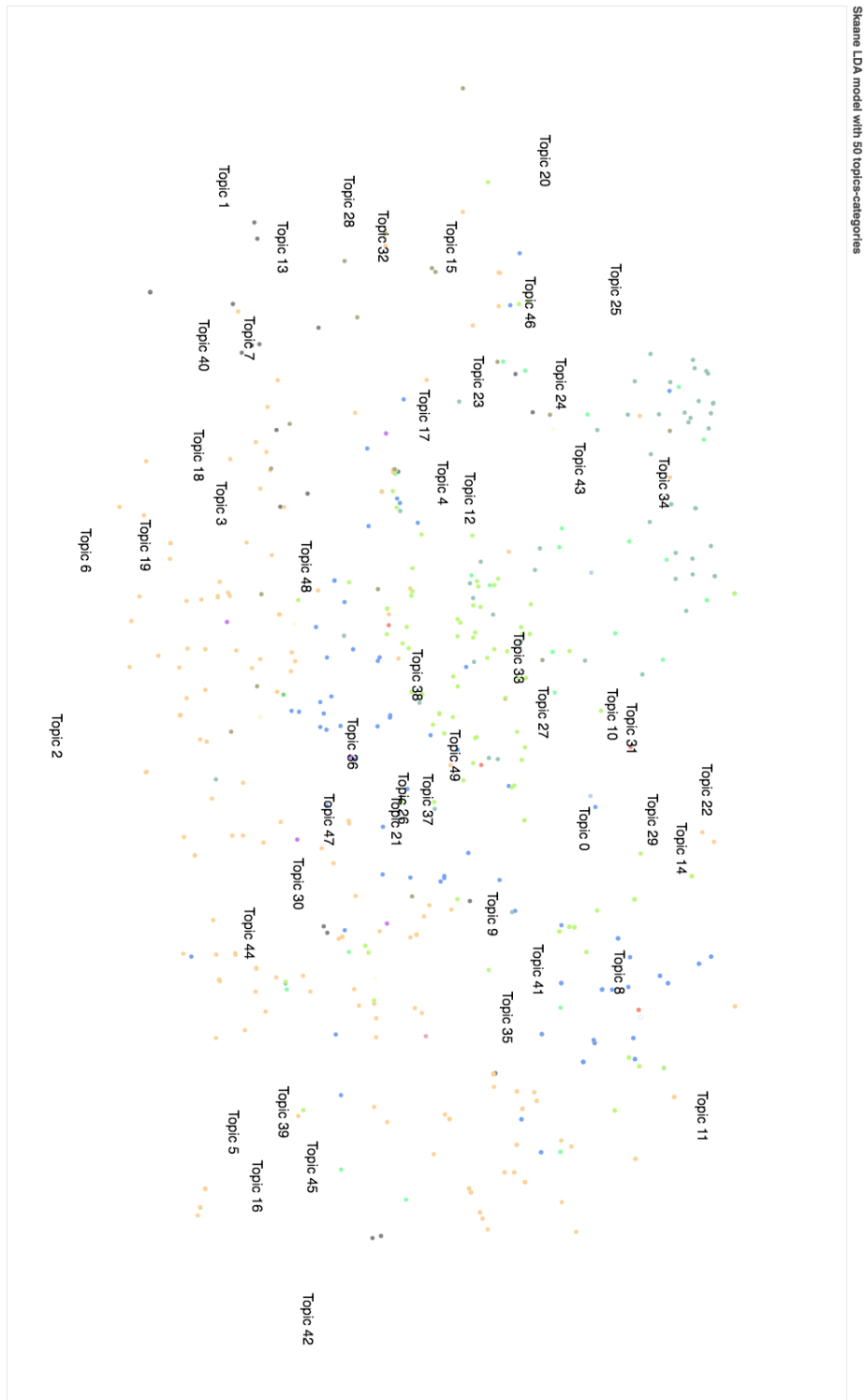
Figure 4.8: The labeled data points plotted in 2D, and colored based on the first label in alphabetical order.
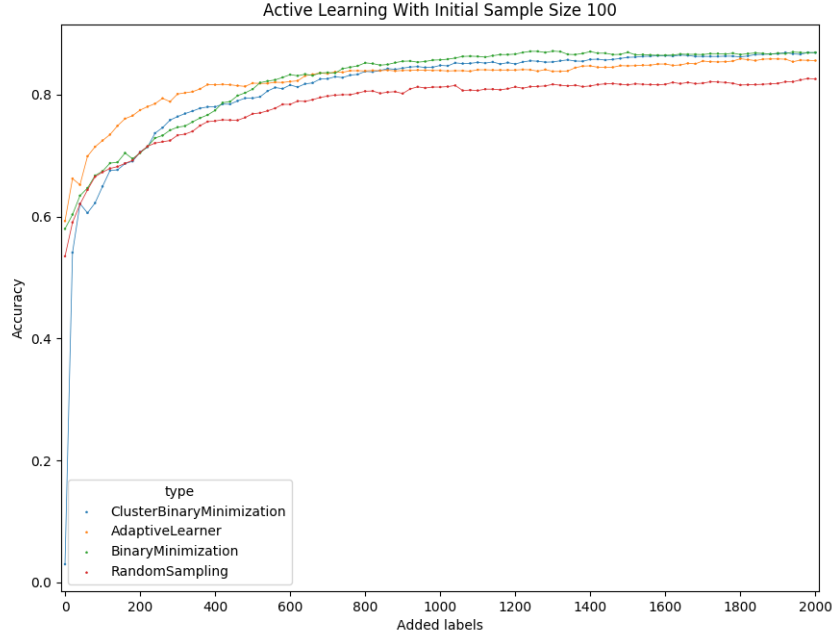
Figure 4.9: Accuracy of the models with initial sample size 25

an significant improvement over labeling at random after round 200 labels. It takes sampling at random approximately 1600 labels to get to an accuracy of 78%. Adaptive Active Learning acquires this accuracy after around 300 labels, and BinaryMinimization and MMC after 450.

When it comes to micro F1-Score, the results between the different methods are similar. Adaptive Active Learning performs better than the rest in the beginning, but the other active learning strategies obtain similar results after around 500 labels. One difference from the accuracy evaluation is that Binary Version Space Minimization does not quite surpass the adaptive learner in the same way. The gap to random sampling is smaller as well. The results can be seen in Figure 4.10 to Figure 4.10.

The same applies to micro recall. The one difference is that the Adaptive Active Learner strategy is performing better than the rest for more iterations. It takes approximately 1000 added labels for the BinaryMinimization to obtain similar micro recall. This can be seen in Figure **??**. The random sampling performs considerably worse than the rest.

The micro precision of the models are very similar between all strategies. Algorithms using clustering obtains a higher precision between 200-400 added labels, but not by that much. From 200 added labels and onwards, Adaptive Active Learner is worse than Binary Version Space Minimization.

Another important aspect of the evaluation is the time it takes. This can be seen in Figure **??**. Time is constant for the random sampling, very close to 0. For the Binary Minimization, the time increases a bit every iteration, but stays under 200 seconds for the first samples. MMC gets similar results, but takes a bit longer. The Adaptive Active Learner takes far longer time than the other methods.

4.4 Evaluating the Label Balance

The last part to evaluate is how the different Active Learning techniques effected the balance of the labels. For the Reuters dataset, how the overall distribution of the labels is can be
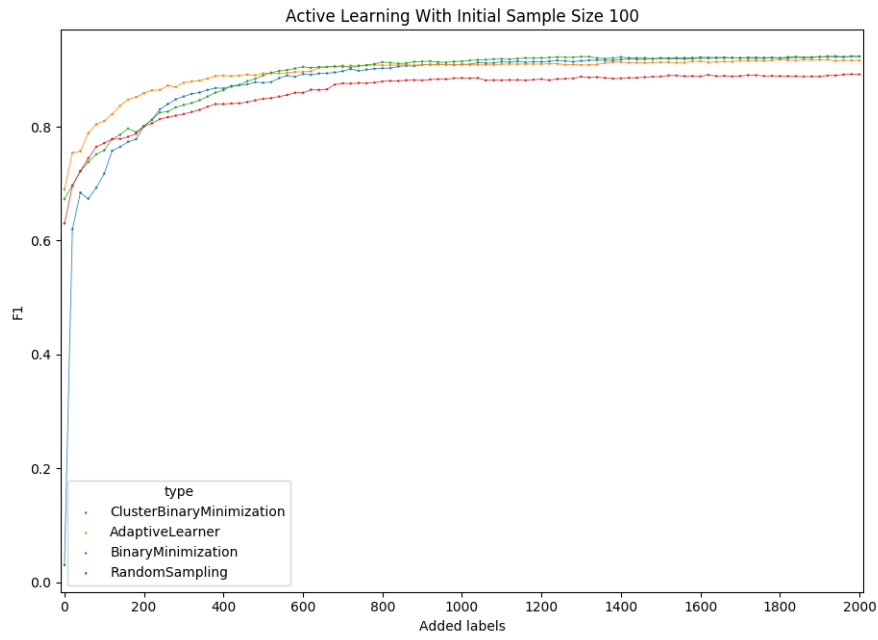
Figure 4.10: Accuracy of the models with initial sample size 25
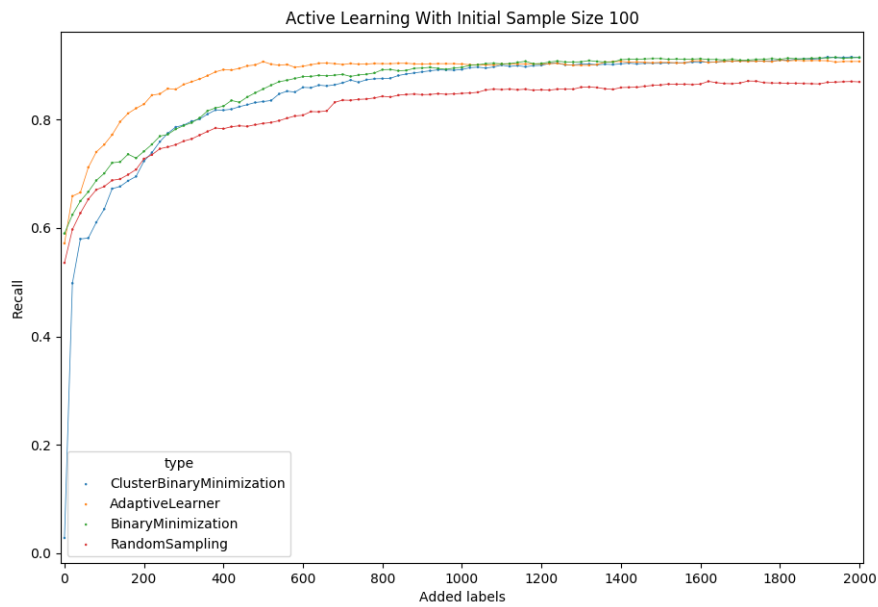

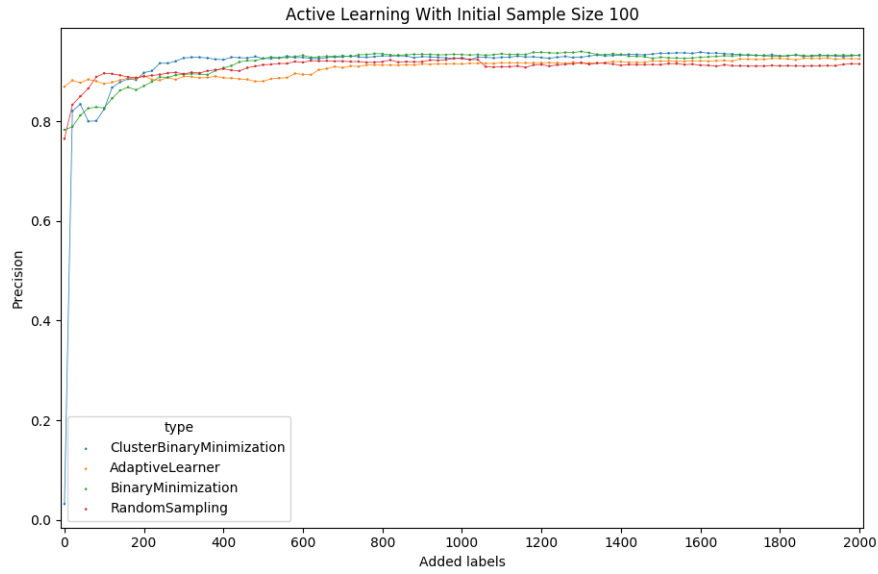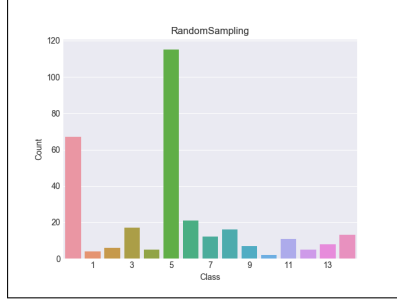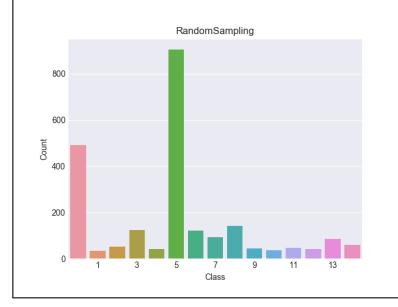Figure 4.11: Micro recall of the models with initial sample size 25

Figure 4.12: Micro precision of the models with initial sample size 25

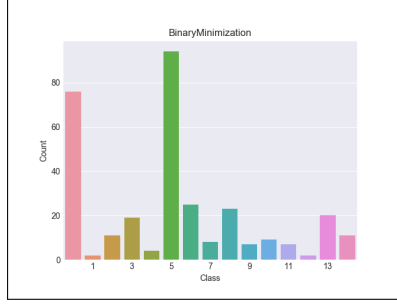Figure 4.13: Time to query a new set of samples by the different strategies

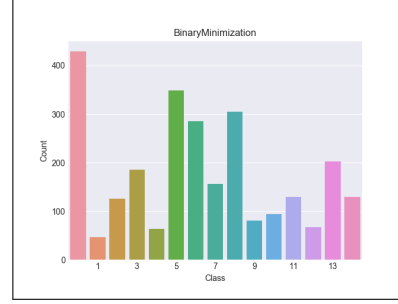(a) The class distribution from random sampling after 200 labels

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

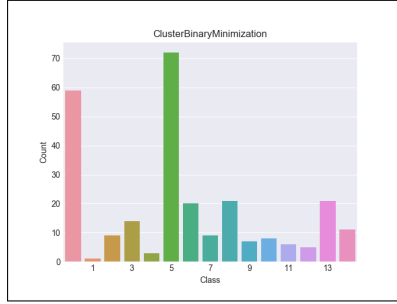Figure 4.14: The distribution of labels after random sampling

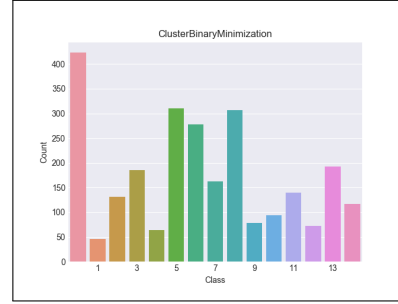(a) The class distribution from Binary Version Space Minimization after 200 labels

(b) The class distribution from Binary Version Space Minimization after 2000 labels

Figure 4.15: The distribution of labels after Binary Version Space Minimization

seen in Figure 3.3. After random sampling, the distribution can be seen in Figure 4.14. The distribution after sampling with the original Binary Version Space Minimization can be seen in Figure 4.15, and with the initial samples taken from clusters in Figure 4.17. The ratio between the biggest and smallest class is here X_1 compared to X_2 for the random sampling. When the initial labeled set was selected from the clusters, the ratio was X. For MMC the distribution can be seen in Figure **??**fig:class-distribution-reuters-mmc, and for Adaptive Active Learning in Figure **??**. The imbalance ratio between the biggest and the smallest class here is X_1 and X_2, respectively. Figure **??**fig:class-distribution-reuters-mmc-clusters and Figure **??**fig:class-distribution-reuters-adaptive-clusters shows the distribution for the same methods, but with the initial samples taken from the clusters.
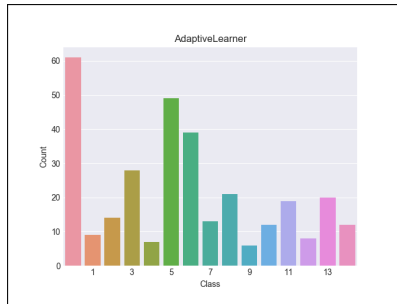
(a) The class distribution from Binary Version Space Minimization with clustering after 200 labels
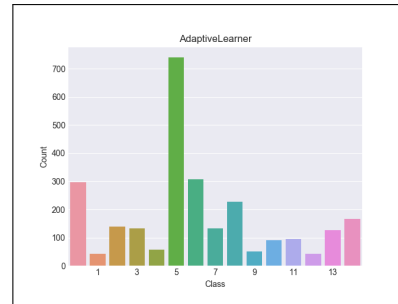
(b) The class distribution from Binary Version Space Minimization with clusterinp after 2000 labels

Figure 4.16: The distribution of labels after Binary Version Space Minimization with clustering

(a) The class distribution from Adaptive Active Learning after 200 labels

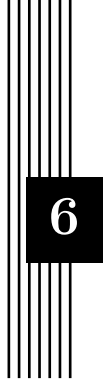(b) The class distribution from Adaptive Active Learning after 2000 labels

Figure 4.17: The distribution of labels after Adaptive Active Learning

5 Discussion

5.1 Results

5.2 Method

6 Conclusion

This chapter contains a summarization of the purpose and the research questions. To what extent has the aim been achieved, and what are the answers to the research questions?

The consequences for the target audience (and possibly for researchers and practitioners) must also be described. There should be a section on future work where ideas for continued work are described. If the conclusion chapter contains such a section, the ideas described therein must be concrete and well thought through.

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: *iinde 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] 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.

[16] Jing Jiang. "Information extraction from text". In: *Mining text data*. Springer, 2012, pp. 11–41.

[17] Thorsten Joachims. "Text categorization with support vector machines: Learning with many relevant features". In: *European conference on machine learning*. Springer. 1998, pp. 137–142.

[18] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. Vol. 344. John Wiley & Sons, 2009.

[19] 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.

[20] Xin Li and Yuhong Guo. "Active Learning with Multi-Label SVM Classification". In: *IJCAI*. 2013, pp. 1479–1485.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. "Classifier chains for multi-label classification". In: *Machine learning* 85.3 (2011), p. 333.

[26] C.J Rijsbergen. *Information Retrieval, 2nd Edition*. 1979.

[27] 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).

[28] 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.

[29] Hinrich Schütze and Craig Silverstein. "Projections for efficient document clustering". In: *ACM SIGIR Forum*. Vol. 31. SI. ACM. 1997, pp. 74–81.

[30] Burr Settles. "Active learning". In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6.1 (2012), pp. 1–114.

[31] 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.

[32] 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.

[33] Simon Tong. *Active learning: theory and applications.* Stanford University, 2001.

[34] 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.

[35] Grigorios Tsoumakas and Ioannis Katakis. "Multi-label classification: An overview". In: *International Journal of Data Warehousing and Mining* 3.3 (2006).

[36] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. "Mining multi-label data". In: *Data mining and knowledge discovery handbook.* Springer, 2009, pp. 667–685.

[37] Vladimir Vapnik. *Statistical learning theory. 1998.* Wiley, New York, 1998.

[38] 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.