Automatic Classification of Mammography Reports

Tim Loyen
Vrije Universiteit Amsterdam
Boelelaan 1105
Amsterdam
t.loyen@student.vu.nl

ABSTRACT

A mammogram is a screening of the human breast using low energy X-rays. Radiologist observe these screenings for the detection of visual indicators of breast cancer. These visual indications are described in a mammography report. Reports are often unstructured and therefore not machine readable. Information therefore has to be retrieved manually, which may be time consuming.

The main purpose of this paper is to research whether a machine can be trained to classify mammography reports. For this experiment a data set containing 17,000 Dutch reports is used. Reports are annotated and contain in some cases multiple labels. These reports originate from the Dutch population screening on breast cancer, IBOB, and were provided by A.J.T.Wanders.

Data is represented through the bag-of-words model. From the data different sizes of N-gram features are extracted. Feature selection is done by both the frequency of a term and two tf-idf approaches. As a classifier Support Vector Machines are used.

The final results of the experiments are a micro averaged F1-measure of 0.896 on the single label data, and a F1-measure of 0.854 on the multi label data. These results suggest that a machine can be trained to classify these reports. Furthermore the data suggest that an increase in training data may also improve the performance.

1. INTRODUCTION

A radiologist is a medical doctor that is specialized in diagnosing and treating diseases using medical imaging techniques. Some examples of such techniques are X-rays, Magnetic Resonance Imaging (MRI) and ultrasound. These techniques can be used for the detection of several types of diseases. A common example of this is breast cancer. Typically for the detection of this disease a mammography is used. A mammography uses low energy X-rays to screen the human breast. The output of the mammography (a mammogram) can then be observed by a radiologist for the detection of cancer.

The task of a radiologist is to scan the mammogram for visual indicators that suggest breast cancer may be present. In the report, the observations made during this session are recorded. Typically a report contains the location, type, density and size of an indication. There are several different methods to record these observations. Digital recording has become more important through the use of the Electronic Health Record (EHR). Depending on the hospital and the radiologist, the findings can be recorded using a keyboard, or through voice recognition software.

Currently when an observation is made by a radiologist, it is recorded and saved as unstructured data. The main consequence of unstructured reporting is that it is not machine readable. Therefore, if information needs to be retrieved from these reports, it has to be done manually, which can be time consuming. Another consequence of unstructured reporting is that observations can be described in several ways. This may in return make it difficult for colleagues to interpret each others transcripts.

One approach to apply structure to these reports is through the use of a protocol. This protocol would hold a set of rules that can structure the way observations are stored by radiologists. Such protocol would however only be effective if it were to be used by a large part of the hospitals in the world, which may make implementation challenging.

In this paper we explore the possibility of having a machine classify these radiology reports. Primarily we are interested in how well a machine can be trained to classify the observations made by radiologist. This research is an initiative by A.J.T.Wanders, who has been a screening radiologist since 1989 [1]. He provided a data set containing 17,000 Dutch mammography reports. These reports originate from the Dutch population screening on breast cancer, IBOB. Each report contains one or more annotations. In total there are 15 different annotations.

In the remainder of the paper, the following content will be discussed. In Section 2, similar studies and common approaches will be discussed. In Section 3 the data set and the methodology will be explained. In Section 4 the results of the experiments will be shown. In Section 5 the results are analyzed and explained. In Section 6 and 7 the conclusion and future work will be discussed.

2. LITERATURE REVIEW

There has already been a significant amount of research done on the classification of radiology reports by machines. One study focused on reports concerning limb fractures [2]. In this study a hundred unstructured text reports were collected from a hospital which were then annotated by experts. Reports could either be annotated as normal (no fracture has been identified) or abnormal (a fracture has been identified in the radiography). Then features were selected for the classification task. For this experiment both standard NLP and domain specific features were collected. Standard NLP features are typically used in almost any domain, such as tokens, token stems, punctuation, bigram and trigrams. They also made use of the SNOMED CT. This is an English terminology service for clinical health information. It may be used to detect clinical terms and to retrieve it synonyms. Features retrieved were then used by a Support Vector Machine (SVM) and a Naïve Bayes Classifier. The SVM reached the highest performance with a F1-measure of 92.31%. The paper concludes that while these results may look promising, further work needs to be conducted to reach the performance of an clinical expert (F1-measure of 98.03%).

The previous study shows that it is possible to classify radiology reports by machines. There are several differences however with the experiment described in this paper. First off all, the study is concerned with a binary classification task, where in paper there are fifteen different classes. Secondly, the dataset is in English, whereas the dataset in this study is in Dutch. The challenge for any Dutch dataset on clinical information is the absence of a terminology service such as SNOWMED CT. Finally the dataset is concerned with reports describing a limb fracture, whereas the dataset in this study is concerned with a mammography.

Another study that focused on mammography experimented with the classification of breast tissue composition [4]. Breast tissue composition is an important component in the evaluation of the breast, however it is rarely reported in coded form, making it challenging for machines to read. Breast tissue composition can be divided into four different categories using the Breast Imaging Reporting and Data System (BI-RAD). There is one extra categories for cases that are ‘unspecified’. For this experiment a relatively large set of reports ($>$ 150,000) were used to construct a set of textual patterns. Using those textual patterns, the classifier correctly predicted 99.8% of the 500 reports.

While the reports in this study were of the same type as the one in ours, there are still some differences. First of all, the dataset contains five classes which still is not comparable to the fifteen in our dataset. Secondly the size of the dataset is significantly bigger then the dataset used in this study. Lastly, every instances in the dataset is assigned to exclusively one class whereas in our dataset there can be instances with two classes.

Furthermore there are some other examples of studies that have researched the classification of radiology reports. One study was concerned with an annotated Spanish dataset of 130,000 reports. Reports either contained pathological findings, or not. [3]. For classification they make use of Radlex, an radiology lexicon, and other NLP techniques. Their final performance was an F1measure of 0.72

The experiment described in this study is known as a document classification task concerning text data. There are multiple classification techniques suited for this problem. One of the simpler techniques is term frequency-inverse document frequency (Tf-idf) [6]. This scoring mechanism uses the frequency of a term in a document and its distribution over the entire dataset to calculate a score. Tf-idf and variations on it have been successfully used in the past for text classification [7][8]

Another well-known approach to the classification of text is the bag of words model. In this technique, documents are converted to vectors, that can then be fed to classifiers. One classifier that performs particularity well with the bag-of-words model and text is the Support Vector Machine(SVM) [9]. SVMs excel at this task because they have a high dimensional input space meaning and because text classification tasks are often lineary seperable [10].

One subject that is not specifically mentioned in the previous studies is the balance of the datasets. As mentioned in the previous section, the dataset used for this study is imbalanced. This means that there are some classes that have a significantly higher frequency than other classes. A wide range of methods have already been studied to deal with this issues. One of the more common approaches is that of under sampling [17]. With undersampling, the amount of instances of the majority class is reduced to balance the ratios between classes.

3. METHODOLOGY

In this section the methods used in the experiments are discussed. First the data set is described. Secondly, several approaches through which data is selected will be discussed. The model through which this data is applied is explained. Finally we discuss how we evaluate the results of the experiments.

3.1 Data set

For this experiment an annotated data set containing 17,000 Dutch reports is used. These reports originate from the Dutch population screening on breast cancer, IBOB, and were provided by A.J.T.Wanders. Each report contains some textual data, and an annotation. Below both elements will be described in more detail.

3.1.1 The reports

The unstructured text was created by radiologists through a speech recognition application. The texts were recorded in the hospitals located in the Netherlands and are therefore written in Dutch. The text consists of 4 to 6 sentences, or on average about 32 words. In these texts, the radiologists report on the observation made in a mammography. Both the primary observation as the secondary information such as location, size and density are reported. An example of the text in a report can be found below:

”Op CC-projectie links lateraal toenemende massa, 12 mm diam. DD compositie/reeel Conclusie: De BI-RADS classificatie voor links is 0: Additionele beeldvorming geindiceerd.”

3.1.2 Annotations

The observations found in mammography are annotated. In this dataset there are five different primary classes. Each class is an indicator that suggest that an abnormality may be found in the mammography. First these five different indicators will be discussed shortly.

- **Massa:** A ‘Mass’ is a space occupying 3D lesion seen in two different projections. If a potential mass is seen in only a single projection it should be called an ’asymmetry’ until its three-dimensionality is confirmed.
- **Calcificatie:** Calcifications are found when an accumulation of calcium is seen on the mammography. Sometimes the presence of calcifications is an indication for malignancy.
- **Architectuurverstoring:** The term architectural distortion is used, when the normal architecture is distorted with no definite mass visible.

- **Asymmetrie:** Asymmetries are findings that represent unilateral deposits of fibroglandulair tissue not conforming to the definition of a mass.
- **Markering:** The last indicator is mark. This is indicator is only used in the Netherlands. It used to ask attention for a specific location on the mammography, without any specifications.

In Table 1 the main classifications and their code can be found. A small part of the instances (399) in the data set did not contain a classification. On recommendation of A.J.T.Wanders these were left out. A challenging aspect of this data set is that in 30.9% of the instances there is a combination of two primary classes. In these instances, more than one indicator was found in the mammography. These combinations are coded by mentioning the code of the most prominent classification first, followed by the number of the second classification. For example if the radiologist observes primarily massa (1MAS) but also calcificatie (2CAL), then the final code will be 1MAS2. It is important to note that 1MAS2 is not the same as 2MAS1. If only one abnormality is found, then the code is followed by its own number (e.g. 1MAS1, 2CAL2 etc.).

Nr.	Name	Code
1	Massa	1MAS
2	Calcificatie	2CAL
3	Architectuurverstoring	3ARC
4	Asymmetrie	4ASY
5	Markering	5INT

Table 1: The primary classes with their codes

In 4.5% of the instances there are multiple classes. For some instances this means that there is a third abnormality found such as in the situation of (1MAS2, 1MAS3).

In Figure 1 the distribution of the primary abnormalities can be found. As described in Section 1, one of the challenges in this study is the imbalance of the data set. It is shown that the MAS and CAL class exceed the other classes greatly.

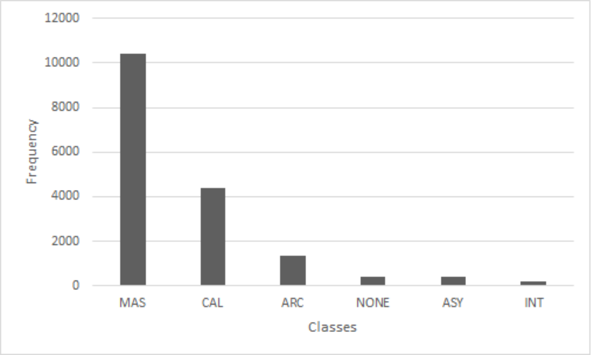

Figure 1: Frequency for the primary classes

In Figure 2 the distribution of all the combinations are shown. Once again the combinations with either MAS or CAL have a greater frequency than the others.

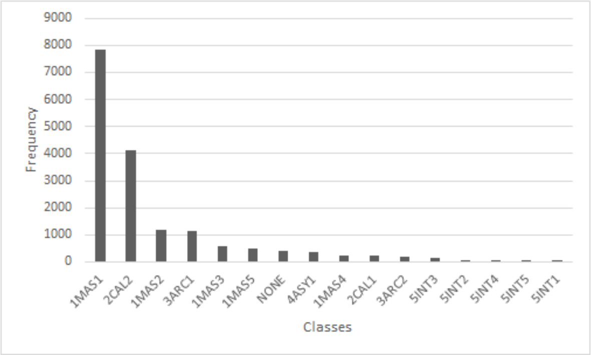

Figure 2: Frequency for the primary classes in combination with the secondary class

3.2 Data set partitioning

Before any experiment was conducted, the data set was split into a training, validation and test set. The validation set is used to experiment with different variables, while the test is used to calculate the final results on. For the partition size the ratio 80/10/10 was chosen. The sizes for both the validation and test set are 1634 instances. The training set got the remaining 11,941 instances. The instances for each set were randomly divided. However, this selection process has been performed multiple times to ensure that each set contained enough instances of every class. In Figure 3 the distribution of the instances over the different partitions can be seen. For most classes however the validation and test set frequencies are too low to be visualized.

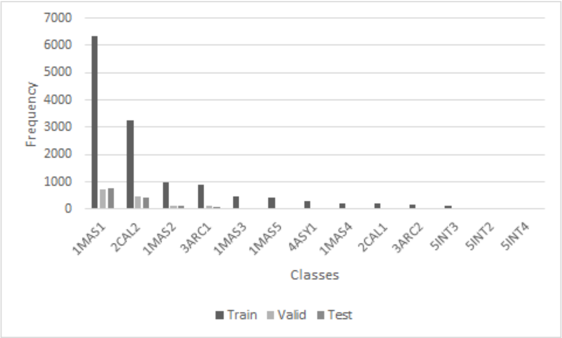

Figure 3: The distribution of all the instances over the training, validation and test set.

3.3 The Bag-of-words model

The first step in a typical machine learning experiment is to acquire features. A feature is an attribute that a machine learning algorithm may use to make its predictions. For the feature selection task in this experiment the bag-of-words(BOW) model is used. The features in a BOW model represent whether a certain term is found in a document. To further explain this concept, an example is given.

Let us take the following two pieces of text that represent our entire data set:

1. After examining the results, patient seems to be stable.
2. After examining the results, patient may need additional testing.

The first step in the BOW model is to create a dictionary. This dictionary contains unique terms that can be found in the data set. Based on these two sentences, the dictionary may contain the following words:

term #	term
1	after
2	examing
3	the
4	results
5	patient
6	seems
7	to

term #	term
8	be
9	stable
10	may
11	need
12	additional
13	testing

Table 2: The dictionary after processing the two example sentences.

In order to train and test a machine learning algorithm, each instance needs to be converted to a vector. A vector can be described as an one dimensional sequence of values. Each value in the vector represents a feature. Vectors are built by using the terms from the dictionary. To illustrate this, let us take the the two sentences from our data set. The first term from the dictionary is *after*. For both instances it is checked whether the term is found in the text. If this is the case the value of the feature is assigned 1, otherwise it is assigned 0. This step is done for every term in the dictionary. The resulting vectors for the two example sentences can be seen in 3

term #	1	2	3	4	5	6	7	8	9	10	11	12	13
vector 1	1	1	1	1	1	1	1	1	1	0	0	0	0
vector 2	1	1	1	1	1	0	0	0	0	1	1	1	1

Table 3: The vectors for the two example sentences.

3.4 Preprocessing

Before features are selected, the textual data of each instances first needs to be processed. The first step is to apply tokenization on the textual data. Tokenization is the task of separating terms from each other in a character sequence. Primarily this means splitting characters based on white spaces. This also includes separating punctuation that may be next to a term. Lastly, any capital characters are transformed to lowercase. For this task we use the tokenizer from the NLTK library[13].

Then stop words can be removed from the set. Stop words are the most common words in a language such as articles (e.g. *the*, *a*). Most likely these words are used in instances of every class, therefore they are not that informative. These words can also have negative effects on the performance of the classification by generalizing classes. Stop words are removed by comparing them to a Dutch stop words data set from the NLTK library[13]. In some approaches stop words can be beneficial, which is explained in the next section.

Another step that helps to narrow down the search space is lemmatization. Lemmatization is the task of converting a word back to its root form. Let us take the following two terms: car and cars. If these terms would be compared directly we would find that they do not match, even though they point to the same object. We therefore chose to normalize them so that they become equal. The Dutch snowball stemmer of the NLTK library[13] was used for this step.

In table 4 each term removal step is shown together with its effect. There are still terms that occur in almost every instance but were not included in our stop word removal data set. These terms can be seen as domain specific stop words. Then a large part of the terms in the set have a low frequency. These terms only occurred in a small part of the instances and are therefor not that relevant for classification. The next section discusses how these terms can be used as features.

Process step	Terms removed	Term count
Initial term count	n/a	6261
Stop term removal	81	6180
Lemmatization	392	5788

Table 4: Terms removed in each step

3.5 Feature selection

In the feature selection step, features are selected based on their relevance towards the prediction task. In the example from the previous section, each unique term was selected to be in the dictionary. However, not all features are equally informative.

3.5.1 N-gram

As explained in subsection 3.3 the features in a BOW model describe whether a certain term was found in the data. Up until now we have only considered using a single term in a feature, which is known as uni-gram. However it is also possible to use the sequence of two or more terms. Again consider the following sentence:

1. After examining the results, patient seems to be stable.

In table 5 three size for n are shown for this sentence.

Unigram	Bigram	Trigram
after	after examining	after examining the
examining	examining the	examining the results
the	the results	the results patient
...	...	...

Table 5: Example of unigrams, bigrams and trigrams

Experiments will be conducted with all three sizes that are shown. Two approaches will be discussed on how features can be ranked and selected for the experiments.

3.5.2 Selection on frequency

The first approach selects terms based on their frequency in the entire dataset. To select these terms, an upper border, and a lower border will be defined. The purpose of the upper border is to exclude domain specific stop words as explained above. The purpose of the lower border is to exclude terms whose frequency is too low for it to be relevant. Optimal borders are selected by means of a grid search approach. The experiments for both the upper and lower border are done separately. This approach has some distinct drawbacks. It only takes the frequency of the term into account, not its distribution over the data set. A term with a high frequency may very well be distributed over only a couple of classes, which would make the term relevant for classification. The next approach is considered to tackle this problem.

3.5.3 Selection on Tf-Idf

The second approach is classification through the numerical statistic tf-idf. Tf-idf stands for term frequency-inverse document frequency and is often used in information retrieval[14]. Tf-idf aims to determine how important a certain term is to a document, given its entire corpus[15]. The term frequency states how often a term occurs in a document. The inverse document frequency then states how many documents (documents in this specific case is the classes) in the entire corpus contain that term. Tf and idf are then combined for one score. A high tf-idf score is reached when the term frequency for a specific document is high, while the document frequency over the entire corpus is low. Let us take for example the term 'fibroadenoom'. It may be that this specific term occurs more often in reports that are classified as Massa than others, thus its occurrence in an instance will increase the chances that the document is classified as Massa.

Two tf-idf approaches are considered. In the first approach, the tf for a term t and a document d is calculated by using the frequency of t in d and dividing it by the maximum frequency of any t in document d. The idf of term t in corpus C is calculated by the amount of documents in C (N) divided by the count of term t in a document d. Finally, tf-idf is calculated by multiplying tf by idf. The equations are shown in Figure 3.

$$tf(t,d) = \frac{f(t,d)}{max\{f(t',d)\}}$$

$$idf(t,C) = log\left(\frac{N}{\{d \subset C : t \subset d\}}\right)$$

$$tf - idf(t,d,D) = tf(t,d) \cdot idf(t,D)$$

Figure 4: The equation for tf-idf where tf is calculated for each class.

In the second approach, tf is adjusted. In the first approach, we calculate tf for each document individually, and take the highest score. Now tf is calculated over the entire corpus.The equations are shown in Figure 4.

$$tf(t,C) = \frac{f(t,C)}{\sum f(t',C)}$$

Figure 5: The equation for tf-idf where tf is calculated over the entire corpus

3.6 Machine learning algorithm

A classifier that is often used in combination with the bag-of-words model and textual data is the Support Vector Machines(SVM)[9]. The SVM places the training data with the different classes into a high dimensional space. In this space, the SVM then attempts to separate the data points from the different classes by constructing a hyper plane. After the training data, new data is presented to the algorithm to measure its performance.

3.7 The classification task

As discussed in Section 3.1, some instances have a primary and a secondary class. In total the dataset contains thirteen unique primary and secondary class combinations, including the classes where the primary and secondary class are the same. In this paper three different classification approaches are explored.

3.7.1 One round classification

In the first approach every instance is classified only once. Each unique combination of classes will become its own separate class. The classifier can thus choose between thirteen different classes for each instance. The approach is simple to execute but does have a drawback. The unique combinations only have a few instances in the training set, making them difficult to classify.

3.7.2 Two round classification

In the second approach, there are two classification rounds. In the first classification round, the classifier will determine for each instance its primary class. Then in the second round the secondary class is classified. An advantage of this approach is that the number of classes on which the classifier is trained is reduced to only five. Each class thus has more instances for training, which may help the minority classes. A drawback however is that misclassifications made in the first round are permanent, and can not be fixed in the second round.

3.7.3 One round classification using probability estimates

In the last approach the probability estimates of the classifier are used. Classifiers such as the SVM return per class scores[18]. In this approach, the classifier is trained on the five primary classes. The primary and secondary classes will then be chosen by taking the first and second highest scores from the probability estimates. Not all instances require a second class however (e.g. 1MAS1, 2MAS2). Therefore needs to be set to determine if the second highest score is considered.

This approach deals well with the drawbacks of the previous approaches. There is only one classification round, and the classifier is trained on only five classes. It should be noted however that this approach assumes that a class is described with the same terms in a primary role as in a secondary role. We may for example find that calcification in 2CAL2 is described with different terms than in 1MAS2. In this case, this approach will not perform as well.

3.8 Class imbalance

As shown in section 3.1 the data set used in this study is unbalanced. Of all instances, 47.5% is of the 1MAS1 class and 24.5% is of the 2CAL2 class. The remaining instances are distributed over the 13 other classes. The problem with unbalanced datasets is that machine learning algorithms tend to only understand the majority class. The algorithm will therefore predict most of the instances to be of the majority class. In this paper, three different approaches will be considered to increase performance on this data set.

3.8.1 Undersampling

A common approach to imbalanced data sets is undersampling[17]. With under sampling, instances from the majority class are removed from the training set. Removing these instances can be beneficial because the bias that the algorithm will have for the majority class may reduce. A drawback of this method however is that a part of the data set will not be used. This may be especially harmful for data sets that are already small, such as the one described in this paper. The undersampling experiments will be primarily focused on the 1MAS1 class. A percentage of instances from this class will be removed randomly. The percentages removed will initially be 25, 50 and 75%.

3.8.2 The C parameter

One of the parameters for the SVM is C. The C parameter stands for the amount of misclassifications that the SVM is allowed to make when constructing the hyper plane [12]. With a high value for C, the SVM will try to avoid making misclassifications on the data set, which may lead to over fitting. It is also possible adjust the C parameter for a specific class. Experiments are done by boosting the C parameter for the minority classes.

3.8.3 Majority class versus the rest

Finally, we consider a majority class versus the rest approach. In this approach, all the instances that are not of the majority class are grouped together. In the first round of classification it will be determined whether an instance is of the majority class or not. Since 47.5% of the instances are of the majority class, there are almost an even amount of instances for each class. In the second classification round, every instance that was not classified as the majority class will be classified to one of the remaining classes.

3.9 Multi-classification

As described in Section 3, a small part of the instances contain multiple annotations. It was decided to leave these instances out of the test set. This choice was made because these instances make the classification process more complex, while there is only a relatively small number of them. It would still be interesting however, to see how well a machine is able to classify these instances. Therefore, two multi classification experiments will be conducted, taking the best parameters of the previous experiments into account.

To run the experiments, the data set will be partitioned again into a training, validation and test set, with the 80/10/10 ratio. Each of these partitions will once again include all thirteen of the active classes. This time however, instances with multiple annotations will not be removed.

From this dataset, two experiments will be conducted. In the first experiment, all the instances with multiple annotations will be removed from the training set. Then thirteen classifiers (one for each class) will be trained on this training set. These classifiers will be trained on whether an instance contains that class or not. Then in the prediction phase, each instance will receive a prediction from all the thirteen classifiers.

In the second experiment, instances with multiple annotations are not removed from the training set. Instead, these instances will be copied for the amount of annotations they hold. Take for example an instance with the annotations 1MAS1 and 1MAS2. The dataset does not specify which annotation is more important, so they should be treated equally. This is done by creating two instances out of this one instance. One instances will have the annotation 1MAS1 and the other will have the annotation 1MAS2, while both instances will have the same data.

3.10 Evaluation

Evaluation of the experiments will be done by analyzing the predictions made by the machine learning algorithm. Primarily the precision and recall measurements will be used. To calculate these measures, the predicted labels are compared to the actual labels. Precision and recall is calculated for each class separately. When comparing these labels, there are four different combinations possible. Below an example is given with the 1MAS1 class.

- True positive(tp): Both the prediction as the actual label are 1MAS1
- False positive(fp): The prediction is 1MAS1 while the actual label is another class.
- False negative(fn): The prediction is another class while the actual label is 1MAS1.
- True negative(tn): Both the prediction as the actual label are another class.

Recall is the fraction of instances classified correctly out of all instances of that class. Precision is the fraction of instances classified correctly out of all positive classifications.

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

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

Precision and recall can be combined using the f1-measure, as defined below.

$$F1 - measure(p) = 2 \cdot \frac{Precision(p) \cdot Recall(p)}{Precision(p) + Recall(p)}$$

The F1-measure is calculated for each class separately. To combine the F1-measures of the different classes, both micro and macro averaged F1-measures are used. The micro averaged F1-measure takes the frequency of each class into account. Classes with more instances (such as the 1MAS1 class) have a higher influence on the final score. This measure is primarily used to get an indication of the overall performance of the system. In the macro averaged F1-measure, each class is weighted equally. This measure can indicate if classes other than the majority class are performing better.

Finally, the multi classification task also need to be evaluated. Because instances can have more than one label, precision and recall as described earlier can not be implemented. Therefore an alternative precision and recall is used, that is calculated over all the results, instead for each class. Let us take all the actual labels T and all the predicted labels P. Precision and recall can then be calculated as:

$$Precision(p) = \frac{\|T \cap P\|}{P}$$

$$Recall(p) = \frac{\|T \cap P\|}{T}$$

The F1-measure then remains the same.

4. RESULTS

In this section, the results of the experiments discussed in section 3 will be shown.

4.1 Majority baseline

In Table 6 the majority baseline is shown. The baseline is produced by assigning to each instance of the test set the majority class (1MAS1). As shown in the Table, the majority class therefore has a optimal recall, but a relatively low precision. Both the micro and macro average f1-measures are calculated as well. Primarily these measures will be used to compare the other experiments to the baseline performance.

Class	Precision	Recall	F1	Frequency
1MAS1	0.457	1.0	0.627	746
1MAS2	0.0	0.0	0.0	106
1MAS3	0.0	0.0	0.0	54
1MAS4	0.0	0.0	0.0	19
1MAS5	0.0	0.0	0.0	42
2CAL1	0.0	0.0	0.0	26
2CAL2	0.0	0.0	0.0	454
3ARC1	0.0	0.0	0.0	115
3ARC2	0.0	0.0	0.0	19
4ASY1	0.0	0.0	0.0	39
5INT2	0.0	0.0	0.0	1
5INT3	0.0	0.0	0.0	11
Micro averaged F1: 0.457				
Macro averaged F1: 0.052				

Table 6: The majority baseline by assigning 1MAS1 to each instance of the test set.

Class	Precision	Recall	F1	Frequency
1MAS1	0.632	1.0	0.775	746
1MAS2	0.0	0.0	0.0	106
1MAS3	0.0	0.0	0.0	54
1MAS4	0.0	0.0	0.0	19
1MAS5	0.0	0.0	0.0	42
2CAL1	0.0	0.0	0.0	26
2CAL2	1.0	1.0	1.0	454
3ARC1	0.0	0.0	0.0	115
3ARC2	0.0	0.0	0.0	19
4ASY1	0.0	0.0	0.0	39
5INT2	0.0	0.0	0.0	1
5INT3	0.0	0.0	0.0	11
Micro averaged F1: 0.734				
Macro averaged F1: 0.15				

Table 7: Baseline performance where the two most frequent classes are always predicted correctly for the single label data

It could be argued however that the majority baseline still is not sufficient. The idea behind the majority baseline is that a under performing classifier would automatically assign each instance to the majority class. However, the 1MAS1 class is not the only class with a significant bigger number of instances. If we were to remove all the instances with 1MAS1 from the data set, then 44.8% of the remaining instances (4,142 out of 9,273) would be of the 2CAL2 class. Thus there may actually be two majority classes. Therefore an alternative baseline was constructed. In this baseline the 1MAS1 and the 2CAL2 instances are always predicted correctly. All the other instances are assigned the 1MAS1 class. The results of this baseline can be found in Table 7

4.2 Selection on frequency

In Table 8 the feature selection experiments based on the frequency of a term in the corpus are shown. In the first column the minimal frequency that a term needs to occur in the dataset is listed. In the second column the amount of features is listed. It is shown that as the term frequency minimum gets higher, the amount of features decreases. In the last two columns the micro and macro average F1-measures are shown. Both the micro and macro average seem to be the highest when terms with a relatively low frequency are used.

Minimal freq.	# Features	Micro avg.	Macro avg.
10	940	0.881	0.594
20	667	0.875	0.577
30	545	0.871	0.578
40	448	0.878	0.589
50	404	0.873	0.572
60	357	0.867	0.564
70	328	0.873	0.572
80	303	0.872	0.551
90	282	0.873	0.554
100	269	0.871	0.56
110	258	0.874	0.571
120	247	0.877	0.58
130	233	0.876	0.56
140	222	0.873	0.561
150	213	0.873	0.557
200	181	0.874	0.543
250	162	0.876	0.564
300	140	0.863	0.546
350	127	0.859	0.51
400	114	0.855	0.498
450	102	0.859	0.517
500	99	0.858	0.517

Table 8: Feature selection based on the minimal frequency of a term

In Table 9 the same experiment is performed but then for the maximal frequency of a term. The best performing minimal frequency (10) is used as a lower boundary, to avoid having too many features. It is shown that removing the most common features negatively impacts the micro and macro average scores.

Maximum freq.	# Features	Micro avg.	Macro avg.
1000	886	0.788	0.411
800	877	0.782	0.411
600	858	0.741	0.368
400	827	0.716	0.348
200	759	0.664	0.269

Table 9: Feature selection based on the maximum frequency of a term

4.3 Selection on Tf-idf

4.3.1 *Tf-idf calculated over the corpus*

In Table 10 the top ten terms are shown ranked by tf-idf over the entire corpus. Tf is here the frequency of a term t in the corpus, and df the amount of classes that contains t. Most terms selected by this approach have a relatively high term and document frequency. None of the terms however are present in every class.

Rank	Term	tf	df
1	klassiek	3128	11
2	birad	2315	11
3	verdacht	4649	12
4	afgrens	883	9
5	groepj	1003	10
6	scherp	1538	11
7	gebied	1494	11
8	calcificaties	1480	11
9	ander	2936	12
10	gladbegrensd	469	8

Table 10: The top ten terms ranked with tf-idf calculated over the entire corpus

In Table 11 the results of the experiments using these terms are shown. In the first column, the amount of terms used is shown. These terms range from rank 1 till rank k. The micro average F1 peaks at 200 features whereas the macro average F1 peaks at 1000 features.

Top k	Micro avg.	Macro avg.
100	0.849	0.452
200	0.868	0.517
300	0.864	0.537
400	0.865	0.562
500	0.867	0.562
600	0.864	0.572
700	0.861	0.569
800	0.857	0.568
900	0.858	0.569
1000	0.859	0.575

Table 11: Experiments with the top k terms with tf-idf calculated over the entire corpus

4.3.2 *Tf-idf calculated for each document*

In Table 12 the top ten terms are shown ranked by tf-idf over each document. Tf is here the frequency of a term t in a class c, df remains the same. The tf-idf for a term t is then calculated for each class where the highest score is used.

Rank	Term	class	tf
1	zer	1MAS4	155
2	klassiek	2CAL2	1755
3	heterog	2CAL1	16
4	calcificaties	2CAL1	110
5	verdacht	1MAS4	206
6	groepj	2CAL2	879
7	biradsclassificatie	5INT3	76
8	zie	5INT3	76
9	brief	5INT3	75
10	mic	3ARC2	8

Table 12: The top ten terms ranked with tf-idf calculated for each document

In Table 13 the results of the experiments using these terms are shown. The results for this tf-idf approach seems to slightly outperform the other approach.

Top k	Micro avg.	Macro avg.
100	0.865	0.524
200	0.875	0.544
300	0.867	0.519
400	0.865	0.552
500	0.866	0.57
600	0.86	0.566
700	0.86	0.562
800	0.861	0.56
900	0.859	0.56
1000	0.86	0.569

Table 13: Experiments with the top k terms with tf-idf calculated for each document

4.4 N-gram

Previously only n-grams of size 1, also called uni-grams were considered. In Table 14 the n-gram experiments are shown with a n of size 2, also called bi-grams. We started with a minimal frequency of 50 to limit the amount of features used. In both micro and macro average, bi-grams seems to perform worst than uni-grams.

Minimal freq.	# Features	Micro avg.	Macro avg.
50	828	0.833	0.498
100	460	0.831	0.496
150	337	0.816	0.445
200	253	0.816	0.409
250	197	0.807	0.392

Table 14: Experiments with a n-gram of size 2

In Table 15 the n-gram experiments are shown with n of size 3, also called tri-grams. In both micro and macro average, bi-grams seems to perform worst than uni-grams and bi-grams.

Minimal freq.	# Features	Micro avg.	Macro avg.
50	708	0.782	0.413
100	344	0.762	0.387
150	228	0.741	0.29
200	147	0.701	0.235
250	111	0.696	0.212

Table 15: Experiments with a n-gram of size 3

4.5 Classification approaches

For the remaining experiments, the best performing features were chosen. These are only uni-gram features with a minimal frequency of 10 and no maximal frequency.

4.5.1 Two round classification

In the previous experiments an one round classification approach was used. In Section 3.7 two other approaches were discussed. In Table 16 the two round classification is shown.

Class	Precision	Recall	F1	Frequency
1MAS1	0.899	0.969	0.933	784
1MAS2	0.706	0.724	0.715	123
1MAS3	0.634	0.553	0.591	47
1MAS4	0.667	0.4	0.5	15
1MAS5	0.765	0.265	0.394	49
2CAL1	0.182	0.105	0.133	19
2CAL2	0.936	0.925	0.93	412
3ARC1	0.807	0.739	0.772	119
3ARC2	0.692	0.45	0.545	20
4ASY1	0.735	0.658	0.694	38
5INT2	0.0	0.0	0.0	1
5INT3	0.0	0.0	0.0	6
Micro averaged F1: 0.861				
Macro averaged F1: 0.529				

Table 16: Two round classification

4.5.2 One round classification using probability estimates

The second approach is an one round approach using the probability estimators of the SVM. The SVM is trained on only the five primary classes. Then the probabilities for each class were used to determine what the (possible) secondary class would be. Before the experiment, the effectiveness of this approach was manually checked. In the test set there are 438 instances where the primary class is different from the secondary class (e.g. 1MAS2). It was found that by using the probability estimates, only in 138 of the instances (32.2%) the secondary class could be found on the second highest probability. From these results was concluded that no further experiments were necessary.

4.6 Class imbalance

4.6.1 Undersampling

In Table 17 the undersample experiments are shown. A percentage of the majority class (1MAS1) is removed from the training set. The micro average decreases slowly as the majority class is reduced. The macro average however seems relatively stable. This can also be seen in Figure 6 where the experiments with 0% and 90% removed are shown.

Removed %	Size training set	Micro avg.	Macro avg.
0	11941	0.881	0.594
10	11361	0.876	0.592
20	10781	0.879	0.605
30	10201	0.876	0.591
40	9621	0.877	0.594
50	9041	0.871	0.606
60	8461	0.869	0.6
70	7881	0.859	0.593
80	7301	0.843	0.577
90	6721	0.822	0.582

Table 17: Undersampling on the 1MAS1 class.

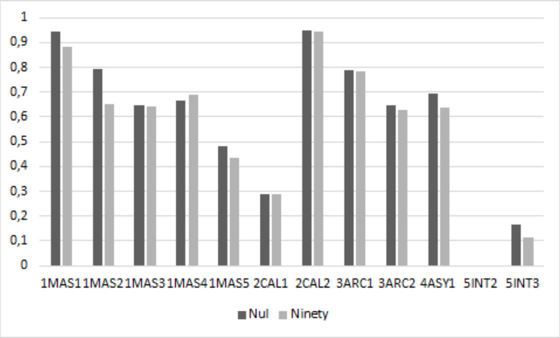

Figure 6: The F1 measure for each class before and after the undersampling.

4.6.2 The C parameter

In Table 18 an experiment is shown where the parameter C (as discussed in Section 3.8.2) is equal for every class. This means that each class can make an even amount of miss classifications when constructing the hyperplane. In Table 18 an experiment is shown where the parameter C is reduced to 0.1 for the majority class 1MAS1. The purpose of this reduction is to reduce the range under which instances will be classified as 1MAS1. This effect is shown in the reduction in recall of the 1MAS1 class. The micro averaged F1 of all the classes is reduced, while the macro averaged F1 stays the same.

Class	Precision	Recall	F1	Frequency
1MAS1	0.934	0.957	0.945	784
1MAS2	0.775	0.813	0.794	123
1MAS3	0.6	0.702	0.647	47
1MAS4	0.667	0.667	0.667	15
1MAS5	0.524	0.449	0.484	49
2CAL1	0.444	0.211	0.286	19
2CAL2	0.945	0.951	0.948	412
3ARC1	0.826	0.756	0.789	119
3ARC2	0.786	0.55	0.647	20
4ASY1	0.703	0.684	0.693	38
5INT2	0.0	0.0	0.0	1
5INT3	0.167	0.167	0.167	6
Micro averaged F1: 0.881				
Macro averaged F1: 0.594				

Table 18: Experiment where C is equal to 1.0 for each class.

Class	Precision	Recall	F1	Frequency
1MAS1	0.967	0.885	0.924	784
1MAS2	0.601	0.846	0.703	123
1MAS3	0.571	0.766	0.654	47
1MAS4	0.714	0.667	0.69	15
1MAS5	0.44	0.449	0.444	49
2CAL1	0.444	0.211	0.286	19
2CAL2	0.94	0.954	0.947	412
3ARC1	0.804	0.756	0.779	119
3ARC2	0.786	0.55	0.647	20
4ASY1	0.638	0.789	0.706	38
5INT2	0.0	0.0	0.0	1
5INT3	0.125	0.333	0.182	6
Micro averaged F1: 0.854				
Macro averaged F1: 0.593				

Table 19: Experiment where C is reduced to 0.1 for the majority class 1MAS1.

4.6.3 Majority class versus the rest

The last approach in this paper to deal with the imbalance of the dataset is to group multiple smaller classes together. For this experiment, every instance that was not 1MAS1 was annotated as 'Other'. Then in the first classification round, each instance is classified as either 1MAS1 or Other. The results of this round are shown in Table 20

Class	Precision	Recall	F1	Frequency
1MAS1	0.942	0.95	0.946	784
Other	0.954	0.946	0.95	850
Micro averaged F1: 0.948				
Macro averaged F1: 0.948				

Table 20: First classification round. All classes except for 1MAS1 are grouped together under the class Other

In the second round of classification, only the instances that were previously classified as Other are considered. These will now be classified into the remaining 12 classes. The results of this round are shown in Table 21

Class	Precision	Recall	F1	Frequency
1MAS2	0.708	0.898	0.792	108
1MAS3	0.632	0.9	0.743	40
1MAS4	0.714	0.667	0.69	15
1MAS5	0.5	0.489	0.494	45
2CAL1	0.444	0.211	0.286	19
2CAL2	0.942	0.954	0.948	411
3ARC1	0.818	0.804	0.811	112
3ARC2	0.733	0.55	0.628	20
4ASY1	0.667	0.828	0.739	29
5INT2	0.0	0.0	0.0	1
5INT3	0.2	0.25	0.222	4
Micro averaged F1: 0.834				
Macro averaged F1: 0.587				

Table 21: Second classification round. Every instance classified as Other is now being classified into one specific class.

Finally the combined results of these two rounds are shown in Table 22

Class	Precision	Recall	F1	Frequency
1MAS1	0.942	0.95	0.946	784
1MAS2	0.708	0.789	0.746	123
1MAS3	0.632	0.766	0.693	47
1MAS4	0.714	0.667	0.69	15
1MAS5	0.5	0.449	0.473	49
2CAL1	0.444	0.211	0.286	19
2CAL2	0.942	0.951	0.946	412
3ARC1	0.818	0.756	0.786	119
3ARC2	0.733	0.55	0.628	20
4ASY1	0.667	0.632	0.649	38
5INT2	0.0	0.0	0.0	1
5INT3	0.2	0.167	0.182	6
Micro averaged F1: 0.876				
Macro averaged F1: 0.591				

Table 22: The results of round 1 and round 2 combined.

4.7 Multi-classification

The multi classification task is done by using the best performing features from the previous experiments into account.

Experiment	Precision	Recall	F1-measure
With double instances	0.887	0.790	0.835
Without double instances	0.877	0.795	0.834

Table 23: Two multi classification experiments. One with and one without double instances in the training set

4.8 Final experiment

These final experiments use the best performing results from previous experiments. The feature selection is done on frequency with a minimal frequency of 10 and no maximum frequency. For the features themselves, only uni-grams are used. The classification is done in only one round, where the classifier was trained on all the thirteen classes. The results of the final single label classification can be found in Table 24.

Class	Precision	Recall	F1	Frequency
1MAS1	0.936	0.962	0.949	770
1MAS2	0.818	0.857	0.837	105
1MAS3	0.635	0.784	0.702	51
1MAS4	0.719	0.793	0.754	29
1MAS5	0.721	0.705	0.713	44
2CAL1	0.833	0.385	0.527	26
2CAL2	0.933	0.935	0.934	429
3ARC1	0.857	0.788	0.821	99
3ARC2	0.688	0.579	0.629	19
4ASY1	0.917	0.688	0.786	48
5INT2	0.0	0.0	0.0	1
5INT3	0.667	0.545	0.6	11
Micro averaged F1: 0.896				
Macro averaged F1: 0.696				

Table 24: The final experiment for the single label instances

The multi label set was performed with the same variables as the single label set. The results of the final multi label classification can be found in Table 25

	F1-Measure
Majority baseline	0,490
Alt. baseline	0,733
Final experiment	0,854

Table 25: The final experiment for the multi label instances

5. DISCUSSION

The main purpose of this paper is to determine how well a machine is able to classify mammography reports. This is not a new domain. In section 2, similar studies have already been discussed.

5.1 Data set

To determine the performance of an experiment, it can be compared to the results of other studies. This turned out to be challenging. The reason for this is primarily because the data set is unique on multiple dimensions. First of all the size of the data set is relatively small. Even though there are 17,000 reports, each reports only contains about 3 to 6 sentences of text. Secondly, because the data set is in Dutch, no clinical lexicon could be used. This put a limitation on the use of more intelligent features. Finally one of the biggest differences with other studies are the annotations. Annotations can be combinations between primary and secondary classes, therefore, the amount of classes becomes fairly large. All of these differences made a comparison with another study too difficult.

5.2 Feature selection

Mainly the features used in this paper were either uni-grams, bi-grams or tri-grams. These features were then selected based on either their frequency in the entire corpus, or their tf-idf score. A baseline experiment was conducted where no selection was done, and simply each term became a feature.

The selection on frequency approach performed the best, but did not perform better than the baseline. The only benefit of this approach was that it reduced the training time of the algorithm greatly. Both tf-idf approaches performed the worst. A possible explanation of this result could be the low frequency of some of the classes. Take for example the term 'bi-rad'. Overall it has one of the highest frequencies in the corpus. It does however not appear in every class (11 out of 13). Because of this, tf-idf classifies this as an useful term. The reason why it does not appear in each class is because some classes do not contain enough instances. Essentially these classes skew the document frequency metric.

Now lets discuss the features themselves. The results from the N-gram experiments show that higher values for N have a negative impact on the performance. There are however some design choices that need to be discussed with these results. First of all, the selection of features on the bigger n-grams was done using the frequency of the features. No experiments with tf-idf were performed, since the algorithm did not seem to perform well. Secondly, the n-gram experiments were done separately from each other. This means that the bi-gram experiment, did not include any uni-grams. It could be argued that n-grams perform the best when the top performing features of the different sizes combined work the best. However, this would require a working feature selection approach.

5.3 Imbalance and size of the dataset

Three approaches have been considered to work around the imbalance problem. First we tried to remove instances from the majority class, lowering the imbalance. Then we tried to tell the classifier to make the majority class less important, by lowering the C value. Lastly we tried to group up all the smaller classes against the majority class. In all three approaches, the overall performance did not improve. Because all these approaches did not improve the performance, perhaps the imbalance of the classes is not a problem. It could be that it is the size of the data or the frequency of some classes.

There are two arguments that support this line of thinking. The first argument is the relative small size of some of the classes. In Section 3.2 we discuss how the two smallest classes were removed because they contained less than three instances. The number three was chosen, because we wanted each partition (training, validation, test) to have at least one instance of each class. However some of the remaining classes are still relatively small (with two instances having $<$ 20 instances). The results suggest that the weak results for these classes was not because of the majority class, it was because the classifier did not have enough instances to train on.

This suggestion that more data could increase the performance is also shown in Figure 7. Here multiple experiments are shown where only a part of the data set is used. Both the micro and macro average show growth when more data is added to the training set. Especially the growth of the macro average is important here, since it proves that it is not just the bigger classes (1MAS1, 2CAL2) that are growing.

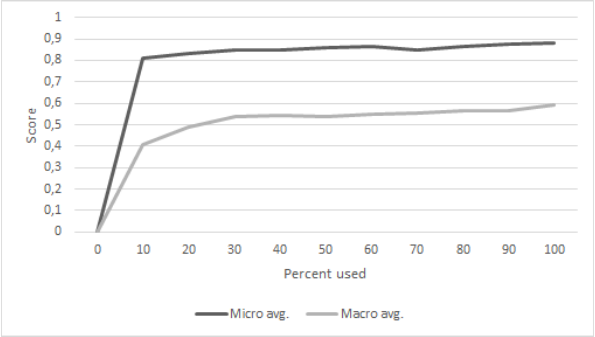

Figure 7: The micro and macro average when training data is increased.

5.4 Evaluation

For evaluation, the metrics recall and precision are used. These metrics are commonly used in document classification, as they offer a better insight in the results than just the accuracy. To combine precision and recall to one value, the F1-measure was used. Primarily it was chosen to use the F1 measure because it was used in other studies reported in Section 2.

The domain of the classification is important however.

Avoiding misclassifications is important in the medical domain. It may therefore not be justified to have recall and precision weighted equally. Precision could be weighted differently, to punish miss classifications harder. The F0.5 measure would be a good alternative since it puts more emphasis on precision.

5.5 Final results

In Section 4.8 the final results for both the single and the multi label classification are shown. As there are no other studies to compare these results with, we primarily use the baseline performance to interpret them. The majority baseline was chosen because almost 50% of the instances in the data set are of the majority class.

In Figure 8 the micro and macro average F1 of the single label classification is compared with both baselines. Here it is shown how poorly the majority baseline performed compared to the actual results. Therefore we also created a stronger baseline, where the two majority classes (1MAS1 and 2CAL2) are always predicted correctly. The micro average baseline performance is increased from 0.457 to 0.734. However the final results from this experiment are still exceeding the baseline performance (0.896 from the final experiment against 0.734 from the baseline). As only two out of the thirteen classes are classified correctly, the macro average F1-measure is not informative.

In Table 25 it is shown how also the final multi label classification experiment is outperforming both baselines.

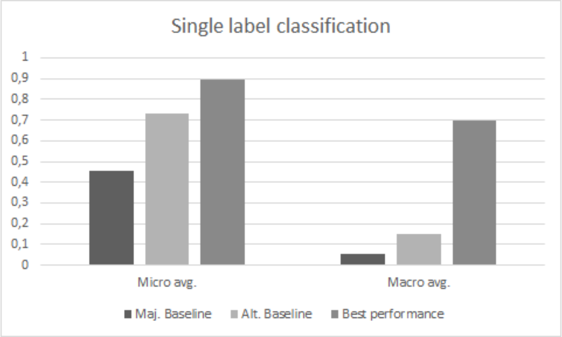

Figure 8: The micro and macro average F1 for the single label classification compared to the baseline

6. CONCLUSION

The main goal of this study was to research whether a machine could be trained to automatically classify unstructured mammography reports. Both the single label as the multi label classification experiments have shown promising results. Furthermore, several indicators have been found that suggest that performance could be improved by increasing the amount of training data. This may especially be useful for classes with the least amount of instances in the current data set.

7. FUTURE WORK

Feature selection is an important task in the classification process. In this paper, multiple approaches have been considered to select informative features. However none of these approaches performed significantly better than the baseline. Therefore, more research could be done in alternative feature selection approaches. Furthermore, only N-gram features are considered in this paper. Therefore other types of features could also be considered.

The results from this paper may be used by A.J.T.Wanders for a project proposal. This project is concerned with the application of the techniques described in this paper within hospitals. Additionally the results may also be used as an example to start similar experiments in other medical domains.

8. REFERENCES

[1] A.J.T.Wanders. 2016. Retrieved 17 July, 2016 from http://www.mammascreening.nl/

[2] Zuccon, G., Wagholikar, A. S., Nguyen, A. N., Butt, L., Chu, K., Martin, S., & Greenslade, J. (2013). Automatic classification of free-text radiology reports to identify limb fractures using machine learning and the snomed ct ontology.AMIA Summits on Translational Science Proceedings, 2013, 300.

[3] Cotik, V., Filippo, D., & Castaño, J. (2014). An Approach for Automatic Classification of Radiology Reports in Spanish. Studies in health technology and informatics, 216, 634-638.

[4] Percha, B., Nassif, H., Lipson, J., Burnside, E., & Rubin, D. (2012). Automatic classification of mammography reports by BI-RADS breast tissue composition class. Journal of the American Medical Informatics Association, 19(5), 913-916.

[5] Liu, Z., Lv, X., Liu, K., & Shi, S. (2010, March). Study on SVM compared with the other text classification methods. In Education Technology and Computer Science (ETCS), 2010 Second International Workshop on (Vol. 1, pp. 219-222). IEEE.

[6] Ullman, J. D., Leskovec, J., & Rajaraman, A. (2011). Mining of Massive Datasets.

[7] Yun-tao, Z., Ling, G., & Yong-cheng, W. (2005). An improved TF-IDF approach for text classification. Journal of Zhejiang University Science A, 6(1), 49-55.

[8] Pandit, S. (2008). On a robust document classification approach using TF-IDF scheme with learned, context-sensitive semantics.

[9] Joachims, T. (2002). Learning to classify text using support vector machines: Methods, theory and algorithms (p. 205). Kluwer Academic Publishers.

[10] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features (pp. 137-142). Springer Berlin Heidelberg.

[11] Drummond, C., & Holte, R. C. (2003, August). C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In Workshop on learning from imbalanced datasets II (Vol. 11).

[12] Veropoulos, K., Campbell, C., & Cristianini, N. (1999). Controlling the sensitivity of sup- port vector machines. Proceedings of the International Joint Conference on AI, 55–60.

[13] S. Bird, E. Klein, E. Loper Natural Language Processing with Python Analyzing Text with the Natural Language Toolkit

[14] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval (Vol. 1, p. 496). Cambridge: Cambridge university press.

[15] Rajaraman, A., & Ullman, J. D. (2012). Mining of massive datasets (Vol. 77). Cambridge: Cambridge University Press.

[16] Rennie, J. D., Shih, L., Teevan, J., & Karger, D. R. (2003, August). Tackling the poor assumptions of naive bayes text classifiers. In ICML (Vol. 3, pp. 616-623).

[17] Liu, X. Y., Wu, J., & Zhou, Z. H. (2009). Exploratory undersampling for class-imbalance learning. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 39(2), 539-550.

[18] Wu, T. F., Lin, C. J., & Weng, R. C. (2004). Probability estimates for multi-class classification by pairwise coupling. Journal of Machine Learning Research, 5(Aug), 975-1005.