Delft University of Technology, Bachelor Seminar of Computer Science and Engineering

Automatic Psychological Text Analysis using Support Vector Machine Classification

Jeongwoo Park[1] , **Willem-Paul Brinkman**[1] , **Merijn Bruijnes**[1]
[1]TU Delft

Abstract

In recent years, there has been an increasing number of patients with mental disorders. A conversational agent is being developed to ensure an easier diagnosis based on the chat between a patient and the agent. The objective of this research is to assess how well Support Vector Machine (SVM) classifies text into its corresponding schema, which are the mental states of the patient. In total, three different classifications have been attempted, Binary, Ordinal, and Per-Questionnaire. The experimental result indicated that SVM is possible to classify 2 out of 7 schema modes, but in general, the performance of SVM was not outperforming with a low f1-score. At the end of the research, SVM was compared to Recurrent Neural Network (RNN) and k-Nearest-Neighbour (kNN) and it turned out that RNN gives the best performance. One of the limitations affecting the result is the quality of the data set. With more correlated labels and a greater size of the data set, improved results can be expected.

1 Introduction

In modern society, many people live with different types of mental disorders. Mood and anxiety disorders compose especially a wide range of mental health problems. Beck's cognitive theory has explained that people with negative cognitive schemas are more vulnerable to depression [1]. Based on this theory, cognitive behavioural therapy has emerged. However, this therapy does not seem to solve problems for chronic and severe patients as the symptoms did not disappear. Jeffrey Young developed schema theory for those patients using the concept "schema mode". Schema modes reflect the moment-to-moment emotional and cognitive state of a person at a given time [2]. Young states that schemas are present in every human being but are manifested more extremely in cases of psychopathology [3]. The theory emphasizes the developmental origins of severe psychopathology [4]. Recently Schema Therapy is gaining popularity for individuals who have different types of mental health and personality problems [2], and also to clinicians and academicians who have started to test both the theoretical assumptions and the clinical effectiveness of this mode [5].

Traditional way of assessing schema modes is to use Short Schema Mode Index (SMI) questionnaire. However, it indeed takes long time and hard to collect the momentary state. Therefore, another approach was contemplated. Instead of answering to the full questionnaire, a conversational agent can be used which shows a shorter adaptive questionnaire [6]. It asks questions related to the events that happened and thus number of questions needed to be answered decreases. In Allaart's research, RASA open-source framework which has a built-in Natural Language Understanding capability was used to automatically analyze the story and then rank the schema modes which are related to that story. However, this text analysis algorithm is lacking accuracy. The text analysis algorithm used in conversational agent barely predicts 2 out of 7 schema modes. With the better classification, the overall performance of the agent can also be improved.

Therefore, three machine learning techniques Support Vector Machine, Recurrent Neural Network and k-Nearest-Neighbour, have been considered to solve the problem, and among all the techniques, this paper mainly discusses Support Vector Machine (SVM). It is a supervised machine learning algorithm which is known to be showing the best results so far in text classification field [7].

The aim of this research project is to implement better text analysis algorithm that classifies into appropriate schemas. This research has focused on two main key points, Text Representation and Classification. Although raw data set has been already prepared, no pre-processing has been applied on it and thus with the original data set it was impossible to classify directly. This study thus handles the possible text pre-processing that allows text classification according to schemas. With this processed data set, SVM classification could be applied and was evaluated using accuracy. In the later stage of the research, SVM was compared to kNN and RNN to find out the best performing technique. The comparison can give an insight on how certain algorithm is competent in the Psychological Text Analysis. With this comparison result, new guideline regarding this research can be suggested, and thus reducing time spent on unlikely results.

Section 2 describes the literature review conducted for this study, followed by section 3 which explains more specific approach and experimental method. Section 4 contains detailed setup and result of experiment conducted. Next, Section 5 reflects on the ethical aspects of this research and discusses

the reproducibility of the used methods. Section 6 discusses limitation in the experiment. Lastly, this paper ends with the summary of research questions and remaining room for improvement.

2 Background

This section goes over the definition of the main research question, "How well can a schema be automatically classified from a text using SVM?". To set up the research methods properly, further literature review should be conducted.

As the research topic is not common in the field of computer science, there were not many examples of similar work that could be directly referred to. Due to this reason, the research topic should be approached in different aspects. This study can be dissected into categories like: Unstructured Text, SVM classification, Text Analysis, Text classification, Text representation, and so on. These key words were used to find out relevant studies.

2.1 Background Information

This research is an extension of Allaart's work [6]. His work aims to create a conversational chatbot that communicates with a patient and then analyzes his/her state. However, this chatbot lacks the text analysis algorithm. This research thus aims to build a text classification algorithm for this chatbot.

Burger has also established a research regarding the same topic, with slightly different schema modes [8]. One key difference in her research was that the labelling of each story was done manually. The core finding from her research is that it is possible to interpret psychological natural language data using a computer algorithm.

Based on these two existing research regarding the same topic, further literature study was conducted to formulate a detailed research question.

Explanation of Schema

Schema mode is a moment-to-moment emotional and cognitive state and coping responses that are active at a given point in time [2]. One of the most commonly used techniques to detect this mode is a mode questionnaire. Schema Therapy is an integrated therapy approach and theoretical framework which is used for the treatment of patients with personality disorders. It targets chronic and characterological disorders rather that acute psychiatric symptoms.

Unstructured Dataset

Before diving into the literature study, knowing the details of the data set used for this study might help understand the general context. The data used for this study was collected by David Allaart [6]. It consists of conversation text and answers to to Schema Mode Inventory (SMI) questionnaires. Conversation text is an extraction from the dialogue between the chatbot and the person. These texts are labelled with the answers to the full questionnaires that the person did after having a conversation with the chatbot with the value between 1 and 6. Therefore, the label reflects on a large time frame, 3 weeks, not a momentary state. This full questionnaire consists of 7 different question sets, each one per schema.

2.2 Text Analysis with SVM

To grasp an idea about schema mode classification, psychological text analysis was studied. Depression, emotional upheavals, and childhood traumas can leave lasting marks on a person's communication which is hard to detect with a naked eye [9]. By analyzing the language of individuals using automated text analysis, these subtle differences can be found that tell people's emotional difficulties. According to [9], to prepare a text for the analysis, a few steps need to be taken. The first step is spelling correction. There are now multiple software programs that automatically correct misspellings such as GNU Aspell. The next step is to extract themes from the text using MEM. MEM uses a statistical procedure known as Principal Components Analysis (PCA). Although this paper briefly handles text preprocessing, it can be assumed that cleaning text and content extraction from text are needed to analyze text.

In addition to psychological text analysis, one of the most related research topics, Sentiment Analysis, was discussed. Sentiment analysis uses texts to determine the attitude/opinion/emotion expressed by a person about a particular topic [10]. Naive Bayesian classification and SVM are some of the most popular supervised learning methods that have been used for sentiment classification [11]. SVM is suitable for a large sample set of the classification, and thus for text classification [12]. It is a state-of-the-art classification algorithm that is known to be successful in many applications [13].

In this work [14], authors proposed a technique to classify the sentiment on the sentiment sand texts for smartphone product review that analyses different data sets used for classification of sentiments and texts. First, data pre-processing was conducted using POS tagging and stop word removal. Then, the score for each sentence is calculated in the document to transform the sentence into a numerical value. Clustering of the document review is based on the TF-IDF measurement. SVM has been used for the sentiment classification part. In the end, the performance was evaluated using Precision, Recall, Accuracy, and F1-Measure.

In addition, feature extraction is one of the important techniques in text classification problems. It reduces the dimension of the vector space, and thus it reduces the complexity of calculation and also prevents overfitting [15]. An efficient way of transforming raw text into vector form is needed such as word embedding which can aid feature extraction.

From these researches, it can be assumed that three steps are required: Data preprocessing, Classification and Evaluation. Another challenge is then how to preprocess data. This research [16] shows the list of preprocessing techniques that can be used. Functionalities that could be applied to our data set are transforming text to lowercase, tokenize data set, word stemming/ lemmatization, and feature selection/weighting.

2.3 Evaluation Metrics

The current data set does not have an even number of data for each class. Using a simple accuracy value as the main evaluation metric is not sufficient to evaluate the performance of the classifier.

In an imbalanced data set, not only is the class distribution is skewed, the misclassification cost is often uneven too. The cost of misclassifying a minority class is much higher than the cost of misclassifying a majority class. Therefore, a different approach to evaluation is needed.

One of the commonly used metrics is the confusion matrix. It is used as a basis for precision and recall and also gives insights into the current status of the model.

F-measure is an evaluation metric that combines precision and recall into a single value. This sentiment analysis work [17] also used Precision, Recall, and ROC which will be described in the next paragraph. Furthermore, sometimes it may occur that a classifier has low accuracy but it has high precision. In a problem where exactness is more important than a high accuracy, looking into precision is very important.

$$Precision = \frac{TP}{(TP + FP)} \tag{1}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{2}$$

$$F - Measure = \frac{Precision * Recall * 2}{(Precision + Recall)} \tag{3}$$

ROC curve displays the tradeoff between the true positive rate and the false positive rate [18]. The advantage of using ROC is that it is shown clearly which region the model is more superior to. The area under the ROC curve, so-called AUC, is often used to represent the performance of the model into a single value.

Lastly, Spearman's rank Correlation can also be used for multiclass classification problems. It is appropriate when the label is a discrete ordinal variable. Intuitively, when the Spearman correlation between two variables is high. It means the values of the two variables are similar. It assesses monotonic relationships between variables.

2.4 Research Questions

After the literature review shown above, three sub-questions were brought up. Splitting the main research question helps sequential approach to the key problem.

1. Which Kernel function of SVM gives the best result?
2. What is the input and output parameters and how should the text be transformed to be used in SVM?
3. What are the differences between the result of three methods, RNN, kNN and SVM?

In order to plug our unstructured data into SVM, text preprocessing was needed. Then, SVM was implemented with Scikit-learn and go through self-evaluation stage. When other classification algorithms are also done with the self-evaluation, the differences between them were compared and analyzed.

3 SVM Schema classification

Classifying text into schemas is a multiclass-multilabel classification problem, meaning there are several target classes and each text can be classified into multiple classes at the same time. The following section explains a theoretical and practical approach to how Support Vector Machine needs to be set up in order to classify text.

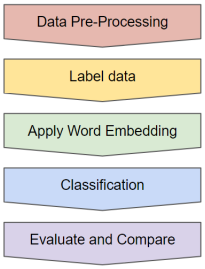

Figure 1: Outline of the Schema Classification

3.1 Data Pre-Processing and Labelling

Data pre-processing is essential to improve the quality of classification. However, this data set contains some small mistakes such as spelling errors, white spaces, unique symbol usage, etc, and also it is not labelled. Labelling is needed to evaluate the classification algorithm so that accuracy can be calculated. Each text was labelled using the questionnaire values from the conversational agent. If the average of each schema mode is above 3.5 or any item of the schema mode is greater than or equal to 5, then it was labelled to that schema mode. Therefore, each text can have multiple labels on it.

Furthermore, in some cases people had noninformative conversation. The standard of 'noninformative' can bring ambiguity, but as there were around 2000 number of short stories, it was difficult to discuss everything together. Thus, following five rules were set for cleaning data set. Whenever the peer was not sure, the peer group decided together about the text.

1. Transform to lower case
2. Remove unnecessary white space
3. Remove comments/ questions unrelated to answering chatbot
4. Remove comments/questions that do not contribute to classifying schema modes
5. Remove general responses (e.g. Ok, Yes, No, Good Bye, Thank you)

After this manual cleaning process, further text preprocessing and techniques are applied to be able to use texts as an input parameter of SVM.

1. Expand contractions
2. Remove stopwords
3. Tokenize the sentence
4. Lemmatization

Contraction is a shorted form of a word, so after expansion, an expression like 'I've' becomes 'I have'.

A stop word is a commonly used word that phrase search has been programmed to ignore. However, there is no official stopwords list that is being used by all NLP tools. Here are

the examples of stopwords from NLTK (Natural Language Tool Kit), e.g. 'but', 'again', 'there', 'about', 'was', so on.

Tokenization is separating one sentence into smaller units called tokens. Tokens can be words, characters, or subwords.

Finally, lemmatization reduces the inflected words properly ensuring that the root word, lemma, belongs to the language. A lemma is a canonical form, dictionary form, or citation form of a set of words [19].

With the combination of these techniques, each story was converted to much simpler and more informative text.

To implement these preprocessing techniques, there were three libraries mainly used. NLTK, Natural Language Tool Kit, contributed to tokenization, lemmatization and removing stopwords [20]. A library called 'contractions' also has been used to expand contraction in the text data. Both libraries are open source.

3.2 Word Embedding with fastText

Word embedding is a method of representing text into a real-valued vector for text analysis so that similar meanings of words end up with closer vectors in the vector space. Each word is mapped to one vector and the vector values are learned using a neural network, and thus this technique can be viewed as a part of deep learning. Word embeddings are easy to work with because they enable efficient computation of word similarities through low-dimensional matrix operations [21].

There are multiple libraries available that support Word-embeddings such as Word2Vec, Doc2Vec, Glove and etc. In this experiment, fastText was chosen with the following reasons.

One of the major drawbacks of Word2Vec and glove is that it cannot deal with unseen words which are not in its corpus. Text data in this experiment contains many informal expressions/ new vocabulary as it is extracted from the recent chatting conversation. Instead of assigning a zero vector to these unknown words, a better approach was needed in order to increase the classifier's performance. FastText can generate vector representation for the words that are not in its corpus by using 'n-gram'. If the value of n is 3 and the word is 'India', then it gives 'in', 'ind', 'ndi', 'di' as the n-gram representations. These representations can be summed to give the vector representation for 'India' [22]. With this concept, fastText can come up with vector forms for unknown words. The training set has vector representations of all its n-grams, so the representation for the unseen word is just the average of vectorized representation of all its constituent n-grams word.

Secondly, there is a pre-trained model available. The advantage of using pre-trained model is that it gives more accurate vector representation as the model is trained on a wide range of data else than the training set. The pre-trained model used for this experiment was trained on Common Crawl and Wikipedia using CBOW with position-weights, in dimension 300, with character n-grams of length 5, a window of size 5 and 10 negatives.

3.3 Theoretical Setup for multi-class and multi-output SVM

Support Vector Machine (SVM) is a supervised machine learning algorithm and is possible to work on both classification and regression problems. This technique is proved to be effective in data mining problems. There are three elements that form the core of SVM: the principle of maximal margin, dual theory, and kernel trick [23]. One of the key advantages of SVM is that it overcame the curse of dimensionality and overfitting problems with the help of the Kernel trick. It is also capable of separating data points with non-linear curve [24].

Kernel function in SVM allows to project original data to a high dimension space [24]. There exist multiple kernel functions.

Linear

$$K\left(x, x_j\right) = x \cdot x^T \tag{4}$$

Polynomial

The output depends on the direction of the two vectors in low dimensional space due to the dot product in kernel [24].

$$K\left(x, x_j\right) = \left(1 + x \cdot x_i^T\right)^d \tag{5}$$

Radial Basis Function

RBF adds a "bump" around each data point.

$$K\left(x, x_j\right) = e^{-\gamma \|x - x_j\|^2} \tag{6}$$

In addition, SVM algorithms are not scaled invariant, so it is highly recommended to scale the data, for instance, by standardizing it to have mean 0 and variance 1 [25].

Furthermore, SVM classification does not support multi-label and multi-class classification natively and requires advanced strategies. One of the classical strategies is to use one-versus-one(1V1). It trains on all possible pairwise binary classifiers and thus results in $\frac{k(k-1)}{2}$ individual classifiers [26]. Each classifier predicts one class and the model with the most votes is the winner.

4 Experimental Setup and Results

Classification was approached with three different methods: Binary classification, Ordinal classification and Per-Questionnaire classification.

4.1 Classification and Evaluation of SVM

SVM was implemented and evaluated with the help of template code provided by Burger Franziska [8]. The main library used to build SVM classifier is Scikit-learn. It has a built-in SVM classifier. Before plugging data into SVM, the vector representation of text is scaled.

Classifier consists of two key stages, train, and prediction. The classifier trains with the training set and then makes a prediction on the test set. The result of this prediction is assessed with diverse metrics. In this experiment, 85% of data was used for training and 15% of data was used for testing.

As our data set is not fully balanced, the experiment included a tuning of the following parameter, 'class_weight',

which was set to 'balanced' or None. It automatically weighs classes like the following:

$$w_j = \frac{n}{kn_j} \tag{7}$$

where w_j is the weight to class j, n is the number of observations, n_j is the number of observations in class j, and k is the total number of classes. w_j is then multiplied to C, penalty for misclassification in SVM, and this calculated value is used for C value for that class.

Binary classification and Per-Questionnaire classification was evaluated using f1-score, confusion matrix, and ROC curve. Spearman's coefficient was chosen for ordinal variable according to [8] as it is easier to interpret relationship between prediction and the actual ordinal value.

4.2 Experiment with Binary Classification

The first classification considered was using a binary label for each schema. Before knowing whether the classifier can detect the degree of each schema, it should be first found out whether the schema can actually detect the presence of each schema from the text.

Setup

As it was introduced in Section 2.1, the original data set used full 67 questionnaire values as labels. Based on the labelling method mentioned in Section 3.1, each schema questionnaire got a True/False label. 67 questionnaire answers could be reduced to in total 7 boolean answers.

The classification was then conducted with conversation text and 7 boolean values as a label per story. Three different kernels, Linear, Polynomial, RBF, were attempted to find out which one gives the best result.

Result

Kernel	f1-score
Linear	0.51653
Polynomial	0.57892
RBF	0.56291

Kernel	Accuracy
Linear	0.024155
Polynomial	0.091787
RBF	0.067632

Table 1: F1-Score and Accuracy of binary classification (class_weight=balanced)

Class	Precision	Recall	F-score	Support
vulnerable	0.27	0.12	0.17	65
angry	0.38	0.75	0.5	83
impulsive	0.07	0.03	0.04	40
happy	0.75	0.89	0.82	156
detached	0.18	0.07	0.1	68
punishing	0.2	0.09	0.12	47
healthy	0.93	0.98	0.95	192
micro avg	0.63	0.63	0.63	651
macro avg	0.4	0.42	0.39	651
weighted avg	0.57	0.63	0.58	651
samples avg	0.66	0.71	0.64	651

Table 2: Classification Report for Binary Classification using Polynomial Kernel (class_weight=balanced)

Looking at Tab 1, it can be found that Accuracy is much lower than f1-score. However, as the data set being used is highly imbalanced, f1-score is the correct indicator to be analyzed.

For all the kernels, it turned out that using 'class_weight = balanced' gives higher f1-score, and among all the kernels, Polynomial gave the highest f1-score. However, the difference of f1-score between Polynomial and RBF is not big, only around 0.015, but the accuracy of Polynomial is higher by 3%.

Thus, Tab 2 shows the specific analysis on Polynomial kernel. Only happy and healthy schema have high precision and recall values. The reason behind this is that those two schemas have highly skewed class distribution so that most of their data is 'True'. According to Fig 6, the model is the best at predicting separation of True Impulsive and False Impulsive as its AUC value is the highest. ROC curve is plotted with False positive rate and True positive rate, meaning increase in False Positive rate can increase AUC value. For Impulsive schema, the classifier is able to detect more numbers of True positives and True negatives than False negatives and False positives. Low F1-Score and High AUC implies that there is a certain threshold for which its score is actually good. As this Impulsive class is imbalanced, it is more appropriate to consider f1-score.

Furthermore, as it was mentioned above, difference between RBF and polynomial is not significant. It is hard to conclude that polynomial will work the best in the binary classification.

4.3 Experiment with Ordinal Classification

As the binary classification was not successful, a new approach to classification was needed. Instead of just figuring out the presence of the schema, this classification was conducted to check whether the classifier works better with multi-labels.

Setup

Rather than using True and False for the y-labels, 1-4 range was chosen for y-labels. The current data set contained 1-6 values for the labels for each schema questionnaire. In order to reflect these numerical values, the average of these values was calculated for each schema. The averaged values were then rounded and mapped to 1-4.

The mapping follows Tab 3 and the following condition: If the answer to one question is greater than or equal to 5 and the mean is smaller than 3.5 then it is mapped to 1.

As Polynomial and RBF kernels gave relatively good results, those were chosen to conduct this experiment.

Average	New label
0-3.5	0
3.5-4	1
4-5	2
5-6	3

Table 3: Mapping of Ordinal Classification

Result

Kernel	Perf
RBF	0.003017
Linear	0.002299
Polynomial	0.009491

Kernel	Perf
RBF	0.005957
Linear	0.014437
Polynomial	0.004598

Table 4: Performance of Ordinal Classification (Left: class_weight=None, Right: class_weight='balanced'

Schema	Spearman
Vulnerable	-0.051687
Angry	-0.043715
Impulsive	0.062300
happy	0.049670
detached	0.138592
Punishing	-0.025186
Healthy	-0.023666

Schema	Spearman
Vulnerable	0.078488
Angry	0.023733
Impulsive	0.003271
happy	0.123908
detached	-0.089540
Punishing	0.073704
Healthy	0.019707

Table 5: Spearman correlation of Ordinal Classification using Kernel: Linear (Left: class_weight = None, Right: class_weight='balanced')

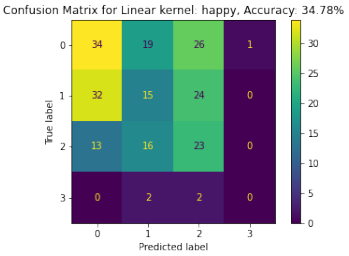

Figure 2: Ordinal Classification: Confusion matrix of Linear Kernel for Happy schema

Tab 16 and Tab 17 shows the result of ordinal classification using Spearman Correlation and a performace metric.

A performance metric was also used to evaluate the model. It is a weighted mean of the spearman correlation for each choice of kernel [8]. The frequencies of schemas in the training set (number of stories with labels > 0 for a given schema/total number of stories) were used as weights.

For both kernels, it is hard to say that setting class_weight to 'balanced' improves the result as it did for binary classification. Polynomial got a better result when the class_weight was set to None and even for RBF, the difference is quite small. Based on the performance score, Linear turned out to be the best kernel for ordinal classification.

Thus, Tab 5 shows specific Spearman Correlation of Linear kernel. Looking at the right table of Tab 5, unlike binary classification, Healthy did not have the highest Spearman correlation. Instead, Vulerable and Punishing got higher value than Healthy, while Happy gives the best result as it did in binary classification. Detached schema even showed a negative value, which indicates the prediction is showing the opposite result relative to the actual result.

Fig 2 also supports why Happy got a relatively high correlation coefficient. Spearman correlation increases when the prediction and the actual ordinal label are similar. The prediction ranged from 0-2 which has the most amount of data, and thus the difference between prediction and the actual result was relatively small.

Combining all these interpretations, it can be concluded that for ordinal classification, setting 'class_weight' to Balanced gives higher performance and Linear kernel gives better result compared to the other two. However, this does not mean that Linear Kernel is outperforming at classification. The positive Spearman correlation for Linear Kernel is still too weak.

4.4 Experiment with Per-Questionnaire Classification

Due to the lacking binary classification, in addition to ordinal classification, this method was also considered. As the classification model used more specific labelling compared to the binary classification, higher performance was expected.

Setup

Unlike ordinal classification, this method takes all the questionnaire values into account. Therefore, in this case, there are 67 labels. Each label range from 1 to 6.

After predicting all the questionnaires, the predicted questionnaire values go through binary labelling. This binary labelled prediction is used for evaluation in a same way as the binary classification.

Same evaluation metrics were used to evaluate Per-Questionnaire classification: Classification Report, F1-Score.

Result

Kernel	f1-score
Linear	0.66268
Polynomial	0.59457
RBF	0.66746

Kernel	Accuracy
Linear	0.024155
Polynomial	0.091787
RBF	0.038647

Table 6: F1-Score and Accuracy of binary classification (class_weight=balanced)

When class_weight is set to 'balanced', all three kernels give better results in terms of f1-score than setting class_weight to None. Moreover, these values are even higher than f1-score from Binary Classification. class_weight='balanced' was selected to be analyzed.

Unlike Binary Classification, RBF gives the best result. The difference between Linear and RBF was minor, so it is hard to conclude that RBF will always give the best result for Per-Questionnaire approach. The noticeable point found in Tab 7 is that Recall values are much higher compared to the binary classification for every schema except Happy and Healthy. Although precision increased except Healthy, it is still low. High recall and low precision indicates that they predict many items as Schema A no matter they actually belong to Schema A.

Although per-questionnaire is not a perfect approach, it is more recommended than binary classification as its f1-score is around 0.9 higher, which is quite significant.

Class	Precision	Recall	F-score	Support
vulnerable	0.34	0.83	0.48	65
angry	0.39	0.76	0.51	83
impulsive	0.2	0.62	0.3	40
happy	0.75	0.88	0.81	156
detached	0.28	0.75	0.41	56
punishing	0.19	0.53	0.28	47
healthy	0.92	0.93	0.93	192
micro avg	0.48	0.82	0.6	639
macro avg	0.44	0.76	0.53	639
weighted avg	0.6	0.82	0.67	639
samples avg	0.51	0.85	0.59	639

Table 7: Classification Report for Per-Questionnaire Classification using RBF kernel

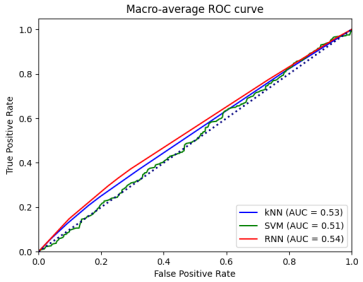

Figure 3: Macro Average ROC-AUC curve

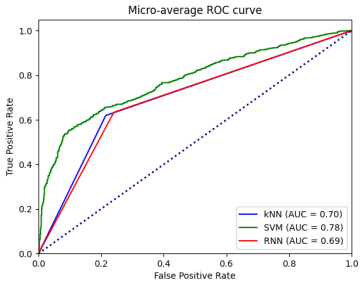

Figure 4: Micro Average ROC-AUC curve

4.5 Interpretation of the results

Each classification approach got the best result with different types of kernels. In general, classifiers gave better results when the class_weight was set to 'balanced'. However, one noticeable point is that RBF was always at least the second rank. Therefore, it is recommended to use RBF kernel if it is not possible to try all types of kernels.

Another finding is that Linear kernel was within the second rank when the classifier used multiple labels for each schema, unlike Binary classification. Therefore, it can be assumed, based on this research, that linear kernel works better under multi-label settings than using the binary label.

Comparing Binary Classification and Per-Questionnaire Classification, when binary output is needed for each schema, using specific questionnaire results for all the schemas like Per-Questionnaire approach is recommended. This can be due to more specific labelling of Per-Questionnaire approach.

The outcome of this research and Allaart's result [6] is similar in the sense that both binary classifiers predicted 2 out of 7 schemas. Comparing Burger's ordinal SVM classifier and this ordinal classifier [8], Burger's classifier was outperforming as it achieved much higher positive correlation for most of the schemas. However, as the data set used for Burger's research and this research and also pre-processing methods differ, it is hard to conclude that the lacking result only depends on the classifier itself. Further investigation is needed to figure out what is the exact cause of the difference.

4.6 Comparison with kNN and RNN

Tab 8 shows a comparison between SVM, kNN and RNN binary classification. According to their F1-score, RNN outperforms compared to SVM and RNN as its average values are higher than other classification models. RNN gives the highest f1-score for all schemas except Happy and Healthy, which kNN exceeds. kNN shows the competent result as it is not far behind RNN. However, SVM does not give an outstanding result to any of the schemas.

According to Fig 3 and Fig 4, SVM, RNN and kNN have similar macro average ROC curve, but SVM has a much higher AUC for micro average. Macro average is just a mean of AUC from all the schemas while micro average can be high if the result of large size schema class is high. All three classifiers have much lower macro and higher micro values. This means that minority schema classes are poorly classified compared to majority classes. In conclusion, RNN gives the best result for binary classification. However, all classifiers are weak against small-size classes.

Schema	SVM	kNN	RNN
Vulnerable	0.27	0.34	0.38
Angry	0.38	0.40	0.48
Impulsive	0.07	0.13	0.17
happy	0.75	0.80	0.77
detached	0.18	0.35	0.36
Punishing	0.2	0.22	0.34
healthy	0.93	0.96	0.95
micro avg	0.63	0.66	0.66
macro avg	0.40	0.46	0.49
weighted avg	0.57	0.62	0.63
samples avg	0.66	0.68	0.66

Table 8: Comparison between SVM, kNN and RNN using F1 Score

Tab 9 shows the comparison of ordinal classification using Spearman Correlation. RNN gives the highest correlation result to most of the schemas except Impulsive and Happy. In general, kNN performs well as it has a postive correlation to all the schemas. This means that kNN tends to make less opposite predictions to the actual result. Meanwhile, SVM and RNN give negative correlations to Detached and Happy respectively. In general, correlations of SVM are lacking compared to kNN and RNN, meaning kNN and RNN are outperforming compared to SVM.

Schema	SVM	kNN	RNN
Vulnerable	0.078	0.13	0.28
Angry	0.023	0.08	0.18
Impulsive	0.0033	0.12	0.042
happy	0.12	0.06	-0.057
detached	-0.090	0.08	0.24
Punishing	0.074	0.09	0.27
Healthy	0.020	0.06	0.09

Table 9: Comparison between SVM, kNN and RNN using Spearman Correlation

5 Responsible Research

This section reflects on ethical aspects of this research and also discusses the reproducibility of the research methods applied to this research.

5.1 Scientific Integrity

This research has referred to multiple related literature. By going through those literature, new idea has been brought up and new concept was learnt. In order to avoid plagiarism during this process, all the inspiration and references were cited properly following IEEE style.

The data set used in this research is collected by David Allaart. His study received ethical approval from the TU Delft University Human Research Ethics Committee. This data was then provided to the peer group via Teams which is secure and prevents data leakage [6].

Furthermore, the research contains the experiment of coding and thus different results were acquired every modification of the code. The result that has been included to this paper was chosen with appropriate reason and has been well explained. Raw data set contained noninformative data which were excluded manually. The reason behind this was well explained in Secion 3 as those are not data that are needed for this classification problem. There was no data manipulation or trimming in this research. Also, to reduce bias in the data set, data has been randomly chosen. However, the original data set was already too biased.

5.2 Reproducibility

This research contains coding and the result acquired from the execution of the code. The link to code is mentioned in Appendix and also the experiment set up was clearly mentioned in Section 4. Furthermore, jupyter notebook used for the implementation can both show executed code and the result. This ensures easier verification of the code. With the help of provided code and the explanation above, it is possible to reproduce the result again.

Data set will not be publicly shared due to privacy issue, but the aim of the algorithm generally classifies text into multiple classes. The algorithm does not depend on the concept of schema for classification, so classification using other data set also needs to work well with the algorithm.

6 Discussion

The goal of this research is to find out how well SVM classifier is good at classifying text into the corresponding schema. The text is obtained from the patients and was collected by David Allaart [6]. However, the result of the SVM classifier is not outstanding. There are several factors that are related to the lacking result of classification.

One of the limitations is labelling. Burger's research, which used manual labelling, gave higher performance compared to the ordinal classification conducted in this research. The labels that are being used for this research is not specific to the text itself but reflect the patient's last three week which are collected using SMI questionnaire. Therefore, in some cases, labelling like figure 5 happens. Due to the limited research time given, it was not possible to manually label all the text without professional psychological knowledge.

Secondly, the size of the data set is small. However, it is very difficult to collect these text data and labels as it requires patient's contribution in terms of their time. Often the performance of the classifier increases with the number of available training data, but the number of available data for this research is lacking.

Moreover, the data set is highly imbalanced. In the case of healthy schema, out of 1375 data (including both training set and test set), only 77 data belongs to 'not healthy'. It becomes hard to classify 'unhealthy' data to 'not healthy'. Other schemas were also mostly imbalanced. This skewed class distribution causes classification much harder. According to Binda's paper [27], with 5869 balanced data set, active learning achieved 70% of accuracy while with David's data, 37% was achieved. This indicates that an improved data set can possibly improve the result of classification. Binda also found out that with a relatively balanced schema, Angry, the active learner creates a more complicated model than other imbalanced schemas.

Lastly, additional feature extraction techniques can be added to the preprocessing part. It can also improve the result as they can reduce problems like overfitting, slow speed and etc.

7 Conclusions

The primary research question for this research was 'How well can a schema be automatically classified from a text using SVM?'. In order to tackle this question, it was broken down into three subquestions:

Which Kernel function of SVM gives the best result?

As stated before, there are three kernel functions that have been used for this experiment, Linear, Poly, and RBF. There was no specific one kernel that outperformed in all three classification methods. Often the difference between the first rank kernel and the second rank kernel is so small that it is hard to conclude the first rank to be the best. However, one of the noticeable fact is that RBF was always within the second rank, unlike other kernels which can be the worst in certain classification method.

What is the input and output parameters and how should the text be transformed to be used in SVM?

Text data has been transformed using data pre-processing and word embedding. Before applying word embedding, data has been preprocessed based on the rules mentioned in Section

Text	is_vulnerable	is_angry	is_impulsive	is_happy	is_detached	is_punishing	is_healthy
hello, billy. i recently had to shut my childcare business for the week. i was upset and angry as it meant i had to let down 6 families and refund their money. me and my boss lost money and it was a sad time for both of us. it was a stressful time and we had to contact a lot of people in order to find out if we could re open.	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE

Figure 5: Unusual labelling of the text

3.1. Lastly, word embedding mapped each story to a vector. The vectors which are the result of all of these processes were used as an input to every classification experiment.

However, output parameters, the labels, were different depending on the experiment. The reason for different labels is to find out the best performance of the classifier. For binary classification, questionnaire values were mapped to true and false. For ordinal classification, the average of questionnaire results per schema was mapped to 0-3 values. Lastly, for multiclass classification, the whole questionnaire values were used as the label.

What are the differences between the result of three methods, RNN, kNN and SVM?

In general, RNN is an outstanding classifier among those three algorithms when RNN manages to classify stories. kNN is also good in both binary and ordinal classification and it also gives positive correlation for all schemas in ordinal classification unlike RNN. SVM is lacking the most especially when it is doing ordinal classification.

8 Future Work

As it was mentioned in Section 6, one of the possible ways to improve the performance of the classifier is to improve the quality of the data set. With a lacking data set, it is hard to find out the problem with the classifier itself. Therefore, more data should be collected, especially the ones belonging to the lacking classes. This can solve the imbalanced data set problem.

The current data set is using the full questionnaire, which was done after chatting, as a label for all the stories that the user wrote. However, it is hard to detect schema in this case. The text itself might contain negative content, but the label can be very positive. This can bring classifier confusion. One of the possible solution is to extend this research with more specific labelling on each text such as manual labelling that Burger did [8]. Another solution can be to collect more story samples from the user and use more stories per label.

Furthermore, in the future, an answer of chatbot can also be included as a part of the data set. Depending on the questions asked by the chatbot, the way patients answer the questions differs.

References

[1] Lisa Hawke and Martin Provencher. Schema theory and schema therapy in mood and anxiety disorders: A review. *Journal of Cognitive Psychotherapy*, 25:257–276, 11 2011.

[2] Jill Lobbestael, Michiel van Vreeswijk, Philip Spinhoven, Erik Schouten, and Arnoud Arntz. Reliability and validity of the short Schema Mode Inventory (SMI). *Behavioural and cognitive psychotherapy*, 38(4):437—-458, 2010.

[3] Samantha A. Masley, David T. Gillanders, Susan G. Simpson, and Morag A. Taylor. A systematic review of the evidence base for schema therapy. *Cognitive Behaviour Therapy*, 41(3):185–202, 2012. PMID: 22074317.

[4] Lisa D. Hawke and Martin D. Provencher. Schema theory and schema therapy in mood and anxiety disorders: A review. *Journal of Cognitive Psychotherapy*, 25(4):257–276, 2011.

[5] Samantha Masley, David Gillanders, Susan Simpson, and Morag Taylor. A systematic review of the evidence base for schema therapy. *Cognitive behaviour therapy*, 41:185–202, 11 2011.

[6] David Allaart. Schema mode assessment through a conversational agent. unpublished thesis, 2021.

[7] Vladimir N. Vapnik. *Introduction: Four Periods in the Research of the Learning Problem*, pages 1–15. Springer New York, New York, NY, 2000.

[8] Burger Franziska. Natural language processing for cognitive therapy: extracting schemas from thought records. 2021.

[9] Ryan Boyd. *Psychological Text Analysis in the Digital Humanities*, pages 161–189. 05 2017.

[10] K. Mouthami, K. Nirmala Devi, and V. Murali Bhaskaran. Sentiment analysis and classification based on textual reviews. In *2013 International Conference on Information Communication and Embedded Systems (ICICES)*, pages 271–276, 2013.

[11] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques, 2002.

[12] Y.H. Chen, Y.F. Zheng, J.F. Pan, and N. Yang. A hybrid text classification method based on k-congener-nearest-neighbors and hypersphere support vector machine. In *2013 International Conference on Information Technology and Applications*, pages 493–497, 2013.

[13] Z. Wang, X. Sun, D. Zhang, and X. Li. An optimal svm-based text classification algorithm. In *2006 Inter-*

national Conference on Machine Learning and Cybernetics, pages 1378–1381, 2006.

[14] Upma Kumari, Arvind K Sharma, and Dinesh Soni. Sentiment analysis of smart phone product review using svm classification technique. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, pages 1469–1474, 2017.

[15] Zhijie Liu, Xueqiang Lv, Kun Liu, and Shuicai Shi. Study on svm compared with the other text classification methods. In *2010 Second International Workshop on Education Technology and Computer Science*, volume 1, pages 219–222, 2010.

[16] Jože Bučar and Janez Povh. A knn based algorithm for text categorization. *Proceedings of the 12th International Symposium on Operational Research in Slovenia, SOR 2013*, pages 367–372, 01 2013.

[17] Mohammad Rezwanul Huq, Ahmad Ali, and Anika Rahman. Sentiment analysis on twitter data using knn and svm. *International Journal of Advanced Computer Science and Applications*, 8(6), 2017.

[18] Cheng Weng and Josiah Poon. A new evaluation measure for imbalanced datasets. volume 87, pages 27–32, 01 2008.

[19] Stemming and lemmatization in python.

[20] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[21] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[22] Chubu. Understanding fasttext:an embedding to look forward to, Sep 2019.

[23] Yingjie Tian, Yong Shi, and Xiaohui Liu. Recent advances on support vector machines research. *Technological and Economic Development of Economy*, 18(1):5–33, 2012.

[24] Arti Patle and Deepak Singh Chouhan. Svm kernel functions for classification. In *2013 International Conference on Advances in Technology and Engineering (ICATE)*, pages 1–9, 2013.

[25] 1.4. support vector machines.

[26] Zhe Wang and Xiangyang Xue. *Multi-Class Support Vector Machine*, pages 23–48. Springer International Publishing, Cham, 2014.

[27] Jahson O'Dwyer Wha Binda. Active learning in reducing human labeling for automatic psychological text classification, 2021.

[28] Jayawant N. Mandrekar. Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, 5(9):1315–1316, 2010.

This appendix shows more tables and figures for each experiment result.

A Implementation Details

Link to the code: https://github.com/jeongwoopark0514/CSE3000-Research-Project

This repository contains all the code for SVM implementation and pre-processing. Further details can be found in README.md.

B Binary Classification

As it can be seen well from the classification report, one of problem when class_weight is set to None is that precision is too low. This is not a good behaviour of classifier as it indicates that the classifier is not able to predict any true positives. Looking into both precision and recall value, we can estimate that the classifier is highly likely to predict to certain class without class_weight=balanced, which is not a desired behaviour.

Class	Precision	Recall	F-score	Support
vulnerable	0.0	0.0	0.0	65
angry	0.0	0.0	0.0	83
impulsive	0.0	0.0	0.0	40
happy	0.75	1.0	0.86	156
detached	0.0	0.0	0.0	68
punishing	0.0	0.0	0.0	47
healthy	0.93	1.0	0.96	192
micro avg	0.83	0.53	0.65	651
macro avg	0.24	0.29	0.26	651
weighted avg	0.45	0.53	0.49	651
avg	0.84	0.65	0.7	651

Table 10: Binary classification with Linear kernel (class_weight=None)

Class	Precision	Recall	F-score	Support
vulnerable	0.15	0.03	0.05	65
angry	0.18	0.04	0.06	83
impulsive	0.09	0.03	0.04	40
happy	0.75	0.94	0.84	156
detached	0.25	0.06	0.1	68
punishing	0.11	0.02	0.04	47
healthy	0.93	0.99	0.96	192
micro avg	0.75	0.53	0.62	651
macro avg	0.35	0.3	0.3	651
weighted avg	0.53	0.53	0.51	651
avg	0.8	0.65	0.68	651

Table 11: Binary classification with Polynomial kernel (class_weight=None)

Class	Precision	Recall	F-score	Support
vulnerable	0.25	0.03	0.05	65
angry	0.08	0.01	0.02	83
impulsive	0.17	0.03	0.04	40
happy	0.75	0.96	0.85	156
detached	0.33	0.06	0.1	68
punishing	0.25	0.02	0.04	47
healthy	0.93	0.99	0.96	192
micro avg	0.78	0.54	0.64	651
macro avg	0.39	0.3	0.29	651
weighted avg	0.55	0.54	0.51	651
avg	0.81	0.65	0.69	651

Table 12: Binary classification with RBF kernel (class_weight=None)

Class	Precision	Recall	F-score	Support
vulnerable	0.29	0.55	0.38	65
angry	0.41	0.52	0.46	83
impulsive	0.19	0.33	0.24	40
happy	0.73	0.46	0.56	156
detached	0.36	0.59	0.45	68
punishing	0.23	0.38	0.29	47
healthy	0.91	0.55	0.69	192
micro avg	0.47	0.5	0.49	651
macro avg	0.45	0.48	0.44	651
weighted avg	0.59	0.5	0.52	651
avg	0.47	0.49	0.44	651

Table 13: Binary classification with Linear kernel (class_weight=Balanced)

Class	Precision	Recall	F-score	Support
vulnerable	0.32	0.48	0.39	65
angry	0.38	0.43	0.41	83
impulsive	0.12	0.12	0.12	40
happy	0.74	0.61	0.67	156
detached	0.3	0.37	0.33	68
punishing	0.2	0.34	0.25	47
healthy	0.92	0.8	0.86	192
micro avg	0.52	0.56	0.54	651
macro avg	0.43	0.45	0.43	651
weighted avg	0.58	0.56	0.56	651
avg	0.57	0.59	0.54	651

Table 14: Binary classification with RBF kernel (class_weight=Balanced)

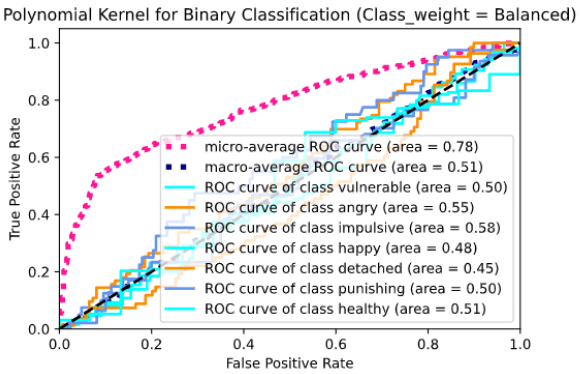

Figure 6: Binary Classification: ROC-AUC of Polynomial Kernel (Class_weight = Balnaced)

Fig 6 shows specific roc curve for each class. It can be seen from the ROC curve that Impulsive schema has the biggest AUC. However, in general, AUC score is around 0.5, meaning no discrimination (i.e., ability to diagnose patients with and without the disease or condition based on the test) [28].

B.1 Confusion Matrix for Polynomial Kernel

Here are the confusion matrices for Polynomial kernel when the class_weight is set to 'Balanced'.

	TP	TN	FP	FN	Accuracy (%)
Vulnerable	8	120	22	57	61.84
Angry	62	23	101	21	41.06
Impulsive	1	153	14	39	74.40
happy	139	5	46	17	69.57
detached	5	116	23	63	58.45
Punishing	4	144	16	43	71.50
Healthy	188	0	15	4	90.82

Table 15: Binary classification: Confusion matrix using Polynomial Kernel and class_weight = 'balanced'

These pessimistic results led to the next stage of classifications, ordinal classification and per-questionnaire classification.

C Ordinal Classification

The following sections shows the classification reports for different kernels, Polynomial and RBF. These two kernels give lower correlation result compared to the linear kernel, and the difference was noticeable as these two involve many negative correlations.

Schema	Spearman
Vulnerable	0.004797
Angry	-0.001477
Impulsive	0.006565
happy	0.072788
detached	0.019651
Punishing	-0.050743
Healthy	0.025854

Schema	Spearman
Vulnerable	0.009380
Angry	-0.008057
Impulsive	-0.002758
happy	0.025170
detached	-0.022593
Punishing	-0.050743
Healthy	0.028532

Table 16: Spearman correlation of Ordinal Classification using Kernel: Polynomial (Left: class_weight = None, Right: class_weight='balanced')

Schema	Spearman
Vulnerable	-0.051687
Angry	-0.043715
Impulsive	0.062300
happy	0.049670
detached	0.138592
Punishing	-0.025186
Healthy	-0.023666

Schema	Spearman
Vulnerable	0.039870
Angry	-0.029001
Impulsive	-0.046340
happy	0.004173
detached	0.069743
Punishing	-0.074156
Healthy	0.037309

Table 17: Spearman correlation of Ordinal Classification using Kernel: RBF (Left: class_weight = None, Right: class_weight='balanced')

D Per-Questionnaire Classification

Tab 18 and Tab 19 show the result of Per-Questionnaire Classification using Polynomial kernel and Linear kernel. It can be seen that Linear kernel has 0 precision for three schemas, while polynomial kernel manages to classify detached and healthy schemas with high precision and high recall values. However, precision and recall for other schemas are too low.

Class	Precision	Recall	F-score	Support
vulnerable	0.15	0.03	0.05	65
angry	0.15	0.02	0.04	83
impulsive	0.1	0.03	0.04	40
happy	0.76	0.95	0.84	156
detached	0.17	0.04	0.06	56
punishing	0.1	0.02	0.04	47
healthy	0.93	0.99	0.96	192
micro avg	0.75	0.54	0.63	639
macro avg	0.34	0.3	0.29	639
weighted avg	0.53	0.54	0.51	639
avg	0.8	0.67	0.69	639

Table 18: Classification Report for Per-Questionnaire Classification using Polynomial kernel

Class	Precision	Recall	F-score	Support
vulnerable	0.0	0.0	0.0	65
angry	0.0	0.0	0.0	83
impulsive	0.33	0.03	0.05	40
happy	0.75	1.0	0.86	156
detached	0.33	0.02	0.03	56
punishing	0.0	0.0	0.0	47
healthy	0.93	1.0	0.96	192
micro avg	0.83	0.55	0.66	639
macro avg	0.34	0.29	0.27	639
weighted avg	0.51	0.55	0.5	639
avg	0.83	0.67	0.71	639

Table 19: Classification Report for Per-Questionnaire Classification using Linear kernel