TARTU UNIVERSITY
Institute of Computer Science
Data Science curriculum

Linda Katariina Grents

Predicting Cognitive Distortions from Reddit Posts by Using Supervised Machine Learning Methods

Master's Thesis (15 ECTS)

Supevisor(s): Kairit Sirts, PhD

Tartu 2022

Predicting Cognitive Distortions from Reddit Posts by Using Supervised Machine Learning Methods

Abstract:

Importance of mental health has gained great attention in modern societies. People have become more open about discussing their thoughts with the public, especially online. One platform that people are using it for is Reddit. The aim of this thesis is to predict cognitive distortions from the texts retrieved from the Anxiety sub-reddit. Cognitive distortions are important to detect as they can potentially have a negative impact on people's lives. Predictions in this work are made by using supervised machine learning methods, such as logistic regression, support vector machine and fasttext (also with pre-trained word vectors). In addition, inter-annotator agreement between annotators is being assessed with Cohen's Kappa and Krippendorff's Alpha. The results show that predicting cognitive distortions from the text is a challenge on its own, since the classifiers were not able to produce satisfactory results. This corresponds to related works where predicting different types of distortions have not given very good results. It is assumed that it would be more reasonable to predict the existence of cognitive distortions from the text rather than predicting different types of distortions, as this prediction shows better results. Predicting the existence of some distortion might be of more help to people suffering from anxiety or depression. It might also be useful to predict only the most prevalent distortions from the text, as some distortions are probably more prevalent than others. It is important to note that major constraint in this work is related to the dataset, as it is relatively small in size and noisy. If there is a need to predict different types of cognitive distortions, it is suggested to use a larger dataset of better quality. However, this remains a challenge on its own in natural language processing and clinical psychology research area.

Keywords:

Cognitive distortions, mental health, AI, NLP, Reddit

CERCS: P176 Artificial Intelligence

Negatiivsete mõttemustrite tuvastamine Redditi postitustest juhendatud masinõppe meetoditega

Lühikokkuvõte:

Vaimse tervise olulisus on ühiskonnas üha enam kõlapinda kinnitamas. Inimesed on muutunud avatumaks ning julgevad oma mõtteid suurema publikuga jagada. Seda on oluliselt lihtsustanud inimeste seotus veebis. Tihtipeale kasutatakse mõtete jagamiseks eri sotsiaalmeedia platvorme, sh Redditit. Selle töö eesmärk on tuvastada Redditi ärevuseteemalise foorumi postitutest kognitiivseid kaldeid ehk negatiivseid mõttemustreid, mis võivad omada negatiivset mõju vaimsele tervisele. Lisaks uuritakse annoteerijate vahelist kooskõla, kasutades Coheni Kappa ning Krippendorffi alfa väärtuseid. Kaldeid tuvastatakse tekstist kasutades juhendatud masinõppe meetodeid nagu logistiline regressioon, tugivektor-masin ning fasttext (samuti eeltreenitud sõnavektoritega). Töö tulemused näitavad sarnaselt varasematele teadustöödele, et kallete tüüpide tuvastamine tekstist on algoritmidele keeruline ülesanne. Tööst selgub, et kalde olemasolu tekstist on algoritmidel lihtsam tuvastada kui kallete tüüpe. Kalde olemasolu tuvastamine võib seejuures olla isegi kasulikum ja efektiivsem kui eri tüüpide tuvastamine, aitamaks depressiivsete kalduvustega või ärevushäirete all kannatavaid inimesi. Samuti võib olla kasulik enam-levinud kallete tuvastamine, sest mõned kalded esinevad tõenäoliselt tihedamini kui teised. Oluline on mainida, et töös kasutatud andmestik on aga üsna väike ja ebatäpne ning eri tüüpide tuvastamiseks on tõenäoliselt vaja suuremamahulist ja parema kvaliteediga andmestikku. See on aga loomuliku keele töötluse ja kliinilise psühholoogia teadusalal omaette väljakutse.

Võtmesõnad:

Negatiivsed mõttemustrid, vaimne tervis, tehisintellekt, loomuliku keele töötlus, Reddit

CERCS: P176 Tehisintellekt

Contents

1. Introduction

During the past years, importance of mental health has been widely discussed in modern societies. Mental well-being has gained equal importance with physical health. While some time ago people were not eager to openly discuss about their issues, concerns and inner thoughts, now, people have become more open about these topics. One of the reasons for this may be widespread access to the internet, which allows people to be more connected. It is quite common for ordinary people to share their inner thoughts with large audience in order to vent their feelings or seek supportive messages from peers. One platform that people are using it for is Reddit and its Anxiety sub-reddit.

As mental health topic is acute in today's society, the author of this thesis decided to investigate this topic further. More specifically, the idea is to make research in clinical psychology using natural language processing methods. The main goal of this thesis is to predict cognitive distortions from Reddit posts derived from the Anxiety sub-reddit. Cognitive distortions are negative thinking patterns that cause people to perceive reality in a distorted way. Anxiety sub-reddit is a good data source to use in this work, as many people with anxiety disorders tend to think in a distorted way. For example, persons suffering from anxiety tend to predict the future, which usually results in catastrophizing the future. Cognitive distortions in this work are predicted by implementing supervised machine learning techniques like logistic regression, support vector machine and Fasttext (also with pre-trained word vectors) on Reddit posts. The data used in this thesis was annotated by Tartu University students and it holds data of 496 Reddit posts along with annotations and metadata. The data was received in ten JSON files from the supervisor of this thesis. In this work, following research questions have been raised:

1) How well have the annotators agreed on labeling cognitive distortions?
2) How well do supervised machine learning techniques predict cognitive distortions from a dataset that is relatively small in size?
3) How do this work results compare to previous works?

This work is divided into multiple sections. Firstly, theoretical background behind cognitive distortions and related works results are being discussed. Secondly, technical background of methods used in this thesis are described. Thirdly, methodology is being discussed followed by the results section. Lastly, results are being discussed followed by the overall conclusion of the work.

The author of this thesis wants to dedicate this work to the people of Ukraine and everyone who has suffered from anxiety or mental health problems due to ongoing war activity in Ukraine. The author of this thesis also wants to thank everyone who has been a part of the writing process – coursemates, friends, head of the Data Science curriculum Jaak Vilo (PhD) and especially the supervisor of this thesis, Kairit Sirts (PhD). Thank you for your support and guidance.

2. Literature Review

2.1 Cognitive Distortions in Essence

Aaron T. Beck described cognitive distortions already in 1963 in his article *'Thinking and Depression. I. Idiosyncratic Content and Cognitive Distortions'* (Beck, 1963). In this study, Beck interviewed 50 psychiatric patients suffering from depression. He noted the following about the depressed patients in his findings: *'A crucial characteristic of the cognitions with this content was that they represented varying degrees of distortion of reality. While some degree of inaccuracy and inconsistency would be expected in the cognitions of any individual, the distinguishing characteristic of the depressed patients was that they showed a systematic error; viz, a bias against themselves.*'. From this finding, it can be concluded that cognitive distortions can be thought of as systematic errors in thinking. Cognitive distortion is described similarly in the American Psychology Association Dictionary of Psychology where it is defined as *'faulty or inaccurate thinking, perception, or belief'*[1]. In addition, Roberts (2017) sees cognitive distortions as toxic automatic thoughts which are in essence untrained impulses resulting from our experiences in life. In general, cognitive distortions can be understood as thinking patterns that cause one to perceive reality in a negative light, which, as a result, affects one's mental well-being and may cause overall down-feeling or depression.

In the literature, cognitive distortions have been categorized differently. There is no definite set of cognitive distortions as different authors tend to categorize cognitive distortions differently and sometimes use different names for the same cognitive distortion. According to Beck (2011), errors in thinking can be categorized into twelve categories: all-or-nothing thinking (or black-and-white, polarized, dichotomous thinking), catastrophizing (or fortune-telling), disqualifying the positive, emotional reasoning, labelling, magnification/minimization, mental filter (or selective abstraction), mind reading, overgeneralization, personalization, *'should'* or *'must'* statements (or imperatives) and tunnel vision. According to Beck & Weishaar (1989) there is also a cognitive distortion called arbitrary inference. In this work, seven cognitive distortions will be predicted: arbitrary inference, black-and-white thinking, catastrophizing, labeling, overgeneralization, personalization and selective abstraction. For the sake of understanding before-mentioned distortions, these will be further explained.

Black-and-white thinking according to Beck (2011) means viewing a situation in two categories. She illustrates this distortion with the following thoughts: *'If I'm not a total success, I'm a failure*' and *'If I can't read the entire chapter, it's not worth reading any of it.'*. Roberts (2017) qualifies black-and-white thinking as polarization and states that it is commonly identified amongst perfectionists. He states that people experiencing this distortion tend to think of themselves either as perfect or trash.

Catastrophizing according to Beck (2011) indicates predicting the future without considering other outcomes which may happen more likely than their prediction. Examples she draws of this distortion are the following: *'I'll flunk out of school'* and *'I'll be so upset, I won't be able to function at all.*'. Cully & Teten (2008) bring another example of this distortion according to which one might think their life is over when one would fail a certain exam.

According to Cully & Teten (2008) **labeling** is seen as *'Giving someone or something a label without finding out more about it/them'*, such as a mother who says her daughter would

[1] APA Dictionary of Psychology

never do something that she disapproved of. According to Jager-Hyman *et al*. (2014) labeling means assigning a derogatory label to oneself or others. Burns (2012) explains labeling as an extreme form of overgeneralization. He explains that labeling oneself is self-defeating and irrational and labeling other people generates hostility. He illustrates his thoughts with an example of a person who labels oneself as a failure after the stock that the person has invested in goes down instead of up, even though the rational way to think here is that the person simply made a mistake and is not a failure because of that.

Overgeneralization according to Roberts (2017) refers to the situation where an isolated event is taken to justify a statement over a larger population. Roberts continues to illustrate this distortion with a situation where someone was swindled by a person wearing suit and tie and a person who heard about it will believe that every person wearing this outfit will be a swindler and, therefore, a dishonest person. In addition, Beck (2011) illustrates this distortion with an example of a person who felt uncomfortable in a meeting and concluded that he/she cannot make any friends with the attendees because of that.

Personalization according to Roberts (2017) is a very damaging error in thinking that is characterized by blaming every bad thing that happens in life on oneself. Roberts illustrates this distortion with an example of a person who is late to a dinner and who will blame oneself because of the quality of food that has been cooked. Cully & Teten (2008) bring another example of this distortion where a mother contemplates what she has done wrong since her daughter has not been talking to her during the day.

Beck (1963) defines **selective abstraction** as follows: *'Selective abstraction refers to the process of focusing on a detail taken out of context, ignoring more salient features of the situation, and conceptualizing the whole experience on the basis of this element.'*. Beck (2011) explains that selective abstraction means focusing only on one negative thing without being able to see the big picture. She illustrates it with an example of a person who thinks he's doing a lousy job because he got a low rating on his evaluation, even though it contained a lot of high ratings as well. Roberts (2017) defines this distortion as filtering and brings an example of a person who has been relocated due to work but instead of seeing it as an opportunity to meet new people, the person focuses only on travelling costs and all obstacles that may arise while adjusting to the new environment.

Arbitrary inference according to American Psychological Association Dictionary of Psychology is defined as *'a cognitive distortion in which a person draws a conclusion that is unrelated to or contradicted by the evidence'*[2]. According to Joshi *et al.* (2021) arbitrary inference can be for example when a person considers vaccines unsafe after seeing media reporting number of deaths after vaccinations. Schuyler (2013) illustrates arbitrary inference with another example about an anxious medical student. According to this example, the student told Schuyler that he was scared that he would fail an exam of a medicine course on his first day of taking the course and would, therefore, have to repeat the course and thought he would fail the exam again which would result him to leave medical school and leave his father furious.

2.2 Cognitive Distortion Prediction

There has not been much research done in cognitive distortion prediction area which makes it extraordinarily interesting to investigate. Despite that, there exist some captivating research work done in this research area.

[2] APA Dictionary of Psychology

Research most closely related to this thesis topic has been made by Sochynskyi (2021) who predicted cognitive distortions from Reddit posts. He used unsupervised learning techniques but also supervised learning techniques such as logistic regression, support vector machine (SVM) and fasttext to make predictions on Reddit data. He predicted the existence of cognitive distortions in text on a binary level but also predicted distortions on a multi-class level. It is seen from the results of his work that fasttext resulted in the highest F-score (0.71) on binary-labelled data and SVM using TF-IDF vectorization technique resulted in the highest F-score (0.23) on multiclass data. Unsupervised learning techniques turned out to be unsuccessful in distinguishing cognitive distortions from text in his work. He also assessed inter-annotator agreement with Cohen's Kappa which resulted in score of 0.569 in the binary setting. It is to be disovered in this thesis how this thesis results can be compared with Sochynskyi's, to find out if the results have been consistent.

Shickel *et al*. (2020) also used machine learning for predicting cognitive distortions but on a different dataset of Sochynskyi's and with some different methods. They used three self-annotated datasets. The first dataset they used was Crowdsourced distortion recollections (CrowdDist) which was derived from a crowdsourcing platform Mechanical Turk and contained 7666 texts labeled into 15 distortions. Second dataset used was mental health therapy Logs (MH) which was derived from an online mental health therapy service TAO Connect and annotated by four senior phsychology students. The aforementioned dataset was divided into two separate ones – MH-C and MH-D – with the first one containing distorted texts with 15 different annotations and second one labeled into binary form (distorted and not distorted). They experimented cognitive distortion prediction with many different algorithms such as logistic regression, support vector machine, random forest, bidirectional encoder representation from transformers etc. It turned out that logistic regression was most outstanding by its performance. They assume it outperformed deep learning techniques because of the small size of the datasets. Binary classification task resulted in a weighted F1-score of 0.88 on the MH-D dataset in their work but failed to classify not distorted texts due to class imbalance. In case of multi-class classification, the weighted F1-score resulted in 0.68 across all texts on CrowDist dataset and the per-distortion F1-scores ranged between 0.55 to 0.77 in their work. The authors believe that the results were affected by short passages of the text and the fact that multiple distortions usually occur together. Classification on the MH-C dataset resulted in worse results which authors justified with the fact that some distortions were unrepresented in the dataset.

Alhaj *et al*. (2022) predicted cognitive distortions from Arabic Twitter using BERTopic. Differently from already mentioned related works, this work was performed on a Twitter dataset and on a text in a different language. The authors note that previous works in the field have given promising results in binary classification setting but in the multi-class setting the work has been rather disappointing. They point out that poor results in previous works may be caused because of shortness of the text being used. They also note that topic modelling in predicting cognitive distortions might be useful, since some cognitive distortions tend to be about some specific topics, e.g they observe that catastrophizing usually relates to relations and academic achievement. Their dataset was labelled by two professional annotators working in psychotherapy field into five different distortion categories: inflexibility, over-generalizing, labeling, emotional reasoning and catastrophizing. They reached quite high agreement score of 0.817 (Cohen's Kappa). The authors of the work decided to pursue classification only with texts that were labelled unanimously. As a result, the final dataset size resulted in 9250 datapoints. They performed pre-processing on the data and split it randomly into 75% training/25% testing. Their results show that the baseline classifiers (decision tree, k-nearest neighbours, support vector

machine, random forest, XGBoost, stacking and bagging) used with Word2Vec could not predict cognitive distortions as well as classifiers that used enriched features. From the findings they conclude that BERTopic algorithm can improve cognitive distortion prediction in multi-class setting.

There is also some interesting work made in this research area by using Linguistic Inquiry and Word Count (LIWC). Simms *et al*. (2017) predicted cognitive distortions on 459 Tumblr posts. In this dataset, 45.1% of the posts were labelled as distorted and 54.9% as not distorted. The authors used LIWC for feature extraction. LIWC is in essence a program that is used in text analysis which can capture proportion of words in text that belong to different psychological categories (Tausczik & Pennebaker, 2010). Simms *et al*. (2017) concluded in their work that RELIEF combined with logistic regression with 10-fold cross-validation gave the best results in their work. For binary classification task they received 73% of accuracy. They conclude that although their results seem promising, they used a rather small dataset and using more sophisticated analytics techniques could provide better results, including achieving lower false negative rate (they achieved 30.4% of false negative rate). They also emphasize on the fact that the ground truth used in their work is not entirely accurate, pointing out that there is some uncertainty with labelling the posts into correct categories.

Similarly to the work of Simms *et al*. (2017), Aureus *et al*. (2021) also used LIWC for feature extraction. Their work focussed on predicting cognitive distortions from Reddit texts associated with COVID-19 pandemic obtained from Subreddit *r/COVID19_support*. The sentences were categorized into 10 cognitive distortions according to the self-developed coding manual by two annotators. Only the intersection of their annotations were used. The final size of the dataset resulted in 586 examples with 50% being annotated as distorted and other half as not distorted. They performed feature extraction and machine learning on the dataset. They found it to be surprising that the distorted texts were less negative than not distorted texts, which, in their words, indicates that cognitive distortions do not always have to contain negative words. They also noted down words occuring mostly in distorted texts, such as higher word count, function words, pronouns and singular first-person pronouns, whileas not distorted texts contained more third person singular pronouns, male references and references to biological processes. As of predicting cognitive distortions with machine learning models, Naive-Bayes using sentiment score resulted in a F1-socre of 0.83 while Linear SVM performed slightly worse. Overall, the authors concluded that psycholinguistic features were significant and help to detect cognitive distortions from text.

Shreevastava & Foltz (2021) predicted cognitive distortions from patient-therapist interactions. In their work, they use a dataset from Kaggle called 'Therapist Q&A'[3] which follows question-answer formatting. In that work, the patient has described its symptoms and thoughts which have been answered by a therapist. These datapoints were annotated by two annotators into eleven categories – into ten different distortions and 'No distortion'. In that work, 3000 samples were annotated which contained 39.2% not distorted datapoints while the remaining was labelled with some sort of distortion. The inter-annotator score (Joint Probability of Agreement) in this work resulted in 33.7% while the score rose up to 61% in the context of binary-labelled data (distorted vs not distorted). As the dataset size is limited, authors decided not to use complex deep learning algorithms for distortion prediction. Instead, they tested four types of features – smooth inverse frequency (SIF), linguistic inquiry and word count features (LIWC), sentence-BERT (S-BERT), parts of

[3] https://www.kaggle.com/datasets/arnmaud/therapist-qa

speech tag embeddings (POS) – with some well-known algorithms: logistic regression, support vector machines, decision trees, k-nearest neighbours and multi-layer perceptron. They found that for binary-labelled classification task, SVM produced the highest F1-score (0.79) and that SIF embeddings performed similarly to BERT embeddings from which they conclude that the word order is not relevant in this task. However, while predicting the presence of cognitive distortions was quite successful, predicting specific distortions from text was not that successful, as none of the algorithms achieved F1-score over 0.30 per distortion in their work. They claim that inter-annotator agreement in their work was a challenge and conclude that it could be low because there is no clear distinction in the literature when it comes to cognitive distortion, as some distortions are grouped into one and there could be multiple labels present for one datapoint entry.

3. Technical Background

Following sub-sections will focus on describing technical background of methods used in this work. Technical background of inter-annotator agreement scores (Cohen's Kappa, Krippendorff's Alpha), text vectorization (TF-IDF), word embeddings and classification algorithms (logistic regression, SVM, fasttext), hyperparameter tuning (grid search) and classification evaluation metrics (precision, recall, accuracy, F1-score) will be described.

3.1 Inter-Annotator Agreement Evaluation Metrics

To better understand how well annotators have agreed with each other while annotating Reddit posts, annotator agreement scores will be calculated. This will bring further insights whether the annotation guideline was well understood and how well available annotations can be trusted. The aim of this sub-section is to explain technical backround behind annotator agreement scores that were used to assess inter-annotator agreement in this thesis.

3.1.1 Cohen's Kappa

Accroding to Cohen (1960), Cohen's Kappa can be calculated in any annotation task that is on nominal scale where two annotators annotated the data. Cohen states that in such annotation task two quantities are relevant:

$$p_0 = proportion\ of\ units\ in\ which\ annotators\ agreed$$

$$p_c = proportion\ of\ units\ in\ which\ agreement\ happened\ by\ chance$$

In this case, $p_0 - p_c$ would indicate the agreement that was reached over chance. Cohen (1960) formulates Kappa score as per following formula:

$$K = \frac{p_0 - p_c}{1 - p_c}$$

Therefore, Kappa value K indicates the level of agreement that was reached over chance. Usually the score ranges between [0:1] where $K = 0$ indicates that there was no agreement between the annotators above chance and $K = 1$ would indicate that the annotators were in complete agreement. By theory, the value could have a negative value as well but this does not occur very often.

To illustrate this, suppose we have two annotators in the annotation task. Let us call them A1 and A2. Both annotators annotated 100 posts into either distorted or not distorted category. Let us say that A1 annotated 75 posts as distorted and 25 posts as not distorted. Let us also say that A2 annotated 50 posts as distorted and 50 posts as not distorted. Supposing that 45 posts were annotated as distorted by both annotators and 30 posts were annotated as not distorted by both annotators, we can calculate proportional agreement level as the following:

$$p_0 = \frac{45 + 30}{100} = 0.75$$

The probability that two annotators both categorized posts randomly as distorted would be the following:

$$\frac{75}{100} \times \frac{50}{100} = 0.375$$

The probability that the annotators both categorized posts randomly as not distorted would be the following:

$$\frac{25}{100} \times \frac{50}{100} = 0.125$$

Therefore, we would get p_c as follows:

$$p_c = 0.375 + 0.125 = 0.5$$

As a result, we can calculate the kappa score using Cohen's formula:

$$K = \frac{0.75 - 0.5}{1 - 0.5} = 0.5$$

According to Landis and Koch (1977), Kappa values can be interpreted as outlined in Table 1:

Table 1. Interpretation of Kappa values according to Landis and Koch (1977).

Kappa value	**Interpretation**
< 0.00	Poor agreement
0.00-0.20	Slight agreement
0.21-0.40	Fair agreement
0.41-0.60	Moderate agreement
0.61-0.80	Substantial agreement
0.81-1.00	Almost perfect agreement

Thus, according to afore-mentioned interpretation table, in this example, the annotators achieved moderate agreement level in the annotation task.

3.1.2 Krippendorff's Alpha

Unlike Cohen's Kappa, Krippendorff's Alpha (KA) can be calculated in situations where a text has multiple labels. According to Krippendorff (2004), KA can be formulated as follows:

$$KA = 1 - \frac{D_0}{D_e} = 1 - \frac{Observed\ disagreement}{Expected\ disagreement}$$

He postulates that when $D_0 = 0$, then $KA = 1$ and when $D_e = D_0$, then $KA = 0$. The values of Krippendorff's Alpha can be interpreted similarly to Cohen's Kappa. Krippendorff (2011) claims that KA of value 0 indicates the absence of reliability, whereas KA of value 1 indicates perfect reliability. He claims that the value usually ranges from $1 \geq KA \leq 0$. KA is rather versatile which means it can be used in different annotation tasks. Krippendorff (2011) states that KA can be calculated in following scenarios:

1. Two or more observers are included in an annotation task;
2. The annotation task may have multiple categories, scale values or measures;
3. The annotation task can use any metric or level of measurement (e.g nominal, ordinal, interval etc.);
4. There can be incomplete or missing data among annotations;
5. The value can be calculated for small or large sample sizes;

The calculation procedure for this metric depends largely of the nature of the data that is at hand. Different calculation procedures are further discussed in Krippendorff (2011).

3.2 Text Vectorization Techniques

In order to perform machine learning operations on textual data, text needs to be vectorized into a numerical form. There are various techniques to vectorize text. In this thesis, Term-Frequency Inverse Document Frequency (TF-IDF) method is used. In Fasttext, word embeddings are used which is why word embeddings are also described. Next sub-paragraphs further explain the background of these methods.

3.2.1 Word Embeddings

Word embeddings are in essence vectors that represent words (Jufrasky & Martin, 2021). The main goal of word embeddings is to create low-dimensional vectors preserving contextual similarity (Millstein, 2019). Word embeddings are mostly used in deep learning models with most popular methods for creating word embeddings being GloVe and Word2Vec (*Ibid*.). Word embeddings are used in many natural language processing tasks, such as information retrieval, machine translation, semantic analysis and dependency parsing (Wang *et al*., 2019).

3.2.2 TF-IDF

Jurafsky & Martin (2021) write that raw frequency is not the best measure when it comes to assessing word associations. They say that ubiquitous words that appear very frequently, such as 'the', 'it' or 'they' are not very informative and are unimportant. They describe that

this problem can be overcome by using TF-IDF weighting. Following description and formulas are based on their work, if not stated differently.

TF-IDF weighting is a product of two terms: term frequency (TF) and inverse document frequency (IDF). TF can be thought of as a frequency of a word *t* appearing in document *d*, which can be formulated as follows:

$$tf_{t,d} = count(t, d)$$

IDF, however, will give higher importance to words that occur only in some documents. Usually, terms that appear in the entire collection are not that helpful. According to IDF, lowest weight will be given to words appearing in all documents. IDF can be formulated as follows:

$$idf_t = \log_{10}\left(\frac{N}{df_t}\right)$$

where *N* denotes the total number of documents in the collection and df_t denotes the number of documents in which the term *t* occurs in. Thus, the TF-IDF weighted value $w_{t,d}$ can be expressed as follows:

$$w_{t,d} = tf_{t,d} \times idf_t$$

The TF-IDF technique indicates which words are most relevant in a document – for example if searching something from a search engine, most relevant documents will be retrieved by using TF-IDF values (Jayaswal, 2020). Jayaswal (2020) explains that TF-IDF results in a vector which has the size of the vocabulary. He states that this technique comes with its own limits. For example, he says that some disadvantages of this technique are its inability to capture word semantics and high cost of computing it on a large vocabulary.

3.3 Supervised Machine Learning Algorithms

Different algorithms can be used to identify patterns or predict something from text, such as sentiments, textual categories and-so-on. In this thesis, supervised machine learning techniques will be used to predict cognitive distortions from the text. Following sub-paragraphs explain technical background behind algorithms used in this thesis – support vector machine, logistic regression and fasttext.

3.3.1 Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning technique which is used in different areas, such as image recognition, hand-writing pattern recognition, protein classification in the medical field and text classification (Gilchrist, 2017). SVM-s perform best in cases where two classes need to be distinguished – it helps to find out the optimal boundary between two classes (Lakshmi T C & Shang, 2021). The mathematical

background of this technique can be difficult to understand for some. To put simply, Vapnik (2000) explains SVM as follows: *'The support vector (SV) machine implements the following idea: It maps the input vectors x into a high-dimensional feature space Z through some nonlinear mapping, chosen a priori. In this space, an optimal separating hyperplane is constructed.*' Graphical representation of SVM by Lakshmi T C & Shang (2021) can be seen from Figure 1.

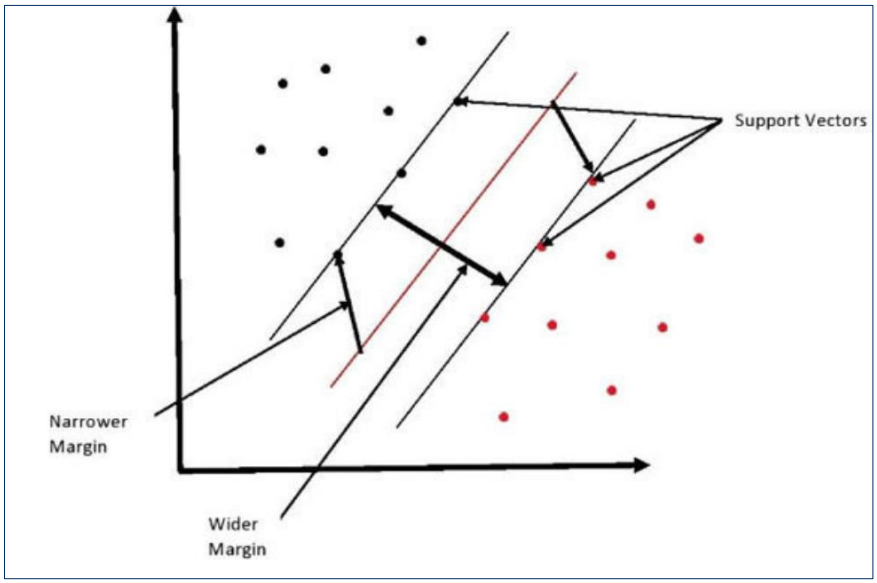


Figure 1. SVM graphic representation according to Lakshmi T C & Shang (2021).

According to Lakshmi T C & Shang (2021), SVM-s are proven to be successful and efficient in text classification tasks. The hyperplane can be thought of as the optimal decision boundary which helps to split two classes, whereas the support vectors are the datapoints most close to both classes (*Ibid*.). The gap between two classes is maximized and if new datapoints are introduced to the model, these will be mapped into the same space and classified into one or the other category based on which side of the gap they will fall into (Marvin, 2021). Marvin continues to explain that there can be multiple hyperplanes and the best hyperplane is the one that maximizes the gap between two classes (called the maximum-margin hyperplane). Gilchrist (2017) notes that SVM-s can be used in linear and non-linear form. Gilchrist adds that SVM has many advantages, such as the effectiveness of it in high-dimensional space and cases where there are less samples than number of dimensions, the fact that it is memory efficient and it performs well with clear margin of separation. However, Gilchrist notes that SVM-s are slow to train on large datasets, the probability estimates are not provided by SVM and, also, it does not perform well with overlapping target classes.

3.3.2 Logistic Regression

Logistic regression is a widely used machine learning technique in multiple fields, including text classification. The following explanation of logistic regression and formulas are based on Hilbe (2015) and Jurafsky & Martin (2021).

Logistic regression model is usually used to predict a binary outcome (0 or 1) based on predictor variable(s). The model has several assumptions, such as that the precitor variables should not be in correlation with each other and that there should be significant relation

between predictor variables and response variable. The model follows Bernoulli probability distribution. Usually probability that the response variable is 1 is predicted and the model results can be interpreted by using oddsratios. Logistic regression model uses maximum likelihood function to predict the probability p that the response variable y is 1. The Bernoulli log-likelihood function is formulated as follows:

$$L(p; y) = \sum_{i=1}^{n} \left\{ y_i \ln\left(\frac{p_i}{1-p_i}\right) + \ln(1-p_i) \right\}$$

The logistic model itself can be formulated as follows:

$$\ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}$$

where x denotes the predictor value and β denotes its coefficient. From this formula we can also notice the odds formula $\frac{p}{1-p}$ which signifies the probability of something occurring divided by the opposite event, that is the probability of something not occurring. The odds can be derived by exponentiating both sides of the logistic regression formula, which is denoted, as a result, as follows:

$$\frac{p_i}{1-p_i} = e^{\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}}$$

By denoting $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}$ as Λ, the probability p that response variable y is 1 can be calculated as follows:

$$p = \frac{1}{1 + e^{-\Lambda}}$$

If there is a need to predict multiple classes, the multinominal logistic regression can be used. It is also called as the softmax logistic regression. In such task, each observation is to be labelled with class k from set of K classes. In multinominal logistic regression if the correct class is denoted as c, then y_c will be set to 1 and other elements of y as 0 (one-hot vector representation). The aim of logistic regression is then to produce an estimate vector $\hat{y}$. Value $\hat{y}_k$ will then be the logistic regression's estimate probability $p(y_k = 1|x)$ for each class k.

3.3.3 Fasttext

Patel (2021) describes fasttext as '*the state-of-the art character-based model*' that is developed by Facebook AI Research. Fasttext can be used in multiple occasions. It is written in the official documentation page of fasttext, that fasttext can be used to learn word

representations and to train text classification models[4]. Fasttext was first mentioned in the article '*Bag of Tricks for Efficient Text Classification*' by Facebook AI Research (Joulin *et al*., 2016), where the authors describe fasttext's model architecture with the following figure (Figure 2):

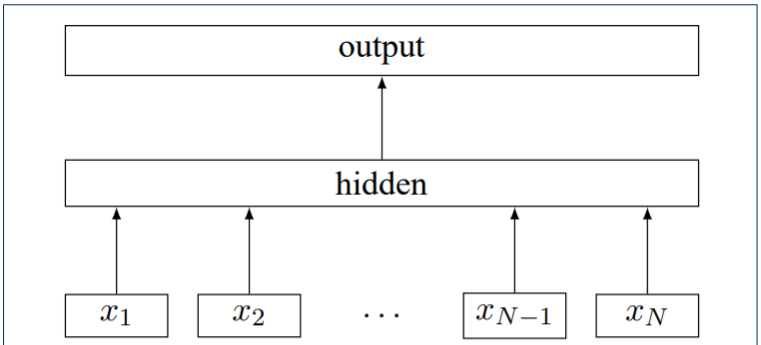


Figure 1: Model architecture of `fastText` for a sentence with N ngram features $x_1, \ldots, x_N$. The features are embedded and averaged to form the hidden variable.

Figure 2. Fasttext model architecture according to Joulin *et al.* (2016).

Joulin *et al*. (2016) describe that the main idea of fasttext is to learn word representations and then average these representations into text representation. They note that the text representation will then be given to the linear classifier. They continue to describe that softmax function will then be used to get probability distribution over classes and negative likelihood will be minimized.

According to Jurafsky & Martin (2021), fasttext is able to tackle out of vocabulary word problem which is for example a disadvantage in case of Word2Vec. They also note that fasttext is able to tackle word sparsity issues with languages that are rich in morphology. They describe the reasons behind that very coherently, as follows: '*Fasttext deals with these problems by using subword models, representing each word as itself plus a bag of constituent n-grams, with special boundary symbols < and > added to each word. For example, with n = 3 the word where would be represented by the sequence plus the character n-grams: <wh, whe, her, ere, re> Then a skipgram embedding is learned for each constituent n-gram, and the word where is represented by the sum of all of the embeddings of its constituent n-grams. Unknown words can then be presented only by the sum of the constituent n-grams.*'. So, as can be understood from this explanation, fasttext is able to break down words into character n-grams and if some word was not seen in the training process, fasttext is able to create word representation from the character n-grams. Also, Gulli *et al*. (2019) mention that character n-grams perform better when it comes to misspelled or rare words.

There is another advantage of fasttext. Grave (2016) writes in the official blog of fasttext that even though usage of deep neural networks is popular in text processing, deep learning models are usually slow to train and test on large datasets. Grave continues to write that fasttext, on the contrary, is able to train much faster by making use of hierarchical classifier. Moreover, Grave states that fasttext is able to produce results of similar accuracy as deep

[4] https://fasttext.cc/docs/en/support.html

learning classifiers. It is to be observed in this thesis, how long fasttext training process will take. There is also another good aspect about fasttext – there is no need to build fasttext vectors by oneself everytime that fasttext is being used for training. In the official fasttext resources page, one can download pre-trained vectors and use these instead.

3.4 Hyperparameter Optimization

Hyperparameter optimization/tuning is an essential part of machine learning. Hyperparameters are something that the model itself is not learning but which, nevertheless, affect the outcome of the training process. There are different ways to implement hyperparameter tuning, out of which grid search is used in this thesis. Grid search is a technique that aims to find optimal parameters from a set of parameters which would provide the best results for a specific algorithm (Sullivan, 2019). In this work, *scikit-learn*'s function *GridSearchCV* was used, which implements cross-validation. According to the *scikit-learn*'s web page about function *GridSearchCV*[5], the function expects a dictionary of parameters to be run on the algorithm and it also allows the user to define a suitable scoring method to find the best parameters. It is argued that, although, grid search is a simple approach, it has several drawbacks, such as the fact that it is computationally intensive, time-consuming and may possibly incurr errors on noisy datasets (Mueller & Massaron, 2016).

3.5 Classification Evaluation Metrics

Machine learning results need to be evaluated in some way to see how well they performed. For this, different metrics can be used. Following explanation and formulas of evaluation metrics are based on Kulkarni *et al.* (2020) and Leung (2022). Confusion matrix can be defined as follows according to Kulkarni *et al*. (2020) (Table 2):

Table 2. Confusion matrix according to Kulkarni *et al*. (2020).

		Predicted	
		Negative	Positive
Actual	Negative	TN	FP
	Positive	FN	TP

where TN stands for True Negative (number of correctly classified negative values), TP stands for True Positive (number of correctly classified positive values), FP stands for False Positive (number of true negative values classified as positive) and FN stands for False Negative (number of true positive values classified as negative).

In many classification tasks accuracy is calculated. Accuracy can be calculated as follows:

$$Accuracy = \frac{TN + TP}{TN + FP + FN + TP}$$

[5] sklearn.model_selection.GridSearchCV — scikit-learn 1.0.2 documentation

However, as for imbalanced datasets accuracy is not the best metric to use, other metrics can be used, such as precision and recall. Precision gives us an idea of how well the model predicted positive outcome and can be calculated as follows:

$$Precision = \frac{TP}{TP + FP}$$

Recall, on the other hand, evaluates model's sensitivity which means that it measures model's strength of predicting a positive outcome. Recall can be expressed as follows:

$$Recall = \frac{TP}{TP + FN}$$

Precision and recall can be combined to formulate a new metric which is oftenly used in classification tasks – F1-score. It takes the harmonic mean of precision and recall and, thus, evaluates tradeoff between correctness and coverage. F1-score can be expressed as follows:

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall}$$

In the multi-class setting, F1-score can be calculated for each class using One-vs-Rest method. However, to get a better glimpse of how well the model performed, micro, macro and weighted F1-score can be calculated. Micro-averaged F1-score is the same as accuracy, thus, expressing how many observations were correctly classified out of all observations. Macro-averaged F1-score takes the arithmetic mean of per-class F1-scores and, thus, treats classes equally by not taking into account their support values (number of observations in a class). Weighted F1-score, on the contrary to macro F1-score, takes the mean of per-class F1-scores by, also, taking into account each class support values. Micro-averaging is best to use with balanced datasets, whileas macro and weighted F1-score are best to use for imbalanced datasets.

4. Methodology

The practical work in this thesis is divided into two sections: preliminary experiment and primary experiment. This is due to the fact that in the beginning of writing this thesis and running experiments, the author of this thesis was not acknowledged to the fact that the annotation guideline that was used by data annotators stated that every text can have only one label. As there are some rows in the dataset which contain multiple labels, the initial objective was to use all the data available and perform binary and multi-label classification. This is called preliminary experiment in this work. However, after preliminary experiment was already executed, the author of this thesis was acknowledged to the fact that every text should in fact have only one label. Therefore, another experiment – primary experiment – was executed where multi-labelled data was eliminated from the dataset and binary and multi-class classification were executed. Both method descriptions will be further discussed in this section. Data and label harmonization techniques are also discussed.

4.1 Data Description

The dataset used in this work originates from Reddit's Anxiety sub-reddit. The posts retrieved from Reddit were annotated by students of Tartu University during the course of Natural Language Processing in 2020 which was held by the supervisor of this thesis Kairit Sirts (PhD). She worked out the annotation guideline which was used by the students in the annotation process.

The data was received from Kairit Sirts in the form of 10 different JSON files. The data was annotated in a way that half of the files included Reddit posts along with annotations and the other half of the files included same Reddit posts with different annotations. In reality, there were ten different annotators who annotated the texts. Each of the annotators annotated maximum of 100 posts. For the sake of simplicity, let us say that the posts were annotated by annotator_1 and annotator_2. Two distinctive datasets by annotator_1 and annotator_2 were joined on unique_id of the posts. After joining the dataset, a complete dataset with 496 posts were retrieved, having labels from annotator_1 and annotator_2. Besides labels given by annotators, unique id of the post and the post itself in textual format, dataset also contained some other columns which were eliminated from the dataset as they were not used in this thesis.

The dataset included some missing (None) and empty ('[]') labels which were not removed as the dataset is relatively small in size. Althogether, there were 5 missing and 1 empty annotations in annotator_1 column and 7 missing and 3 empty annotations in annotator_2 column. Some minor manipulations were made to labels in incorrect formatting. The count of labels per annotator can be seen from Figure 3.

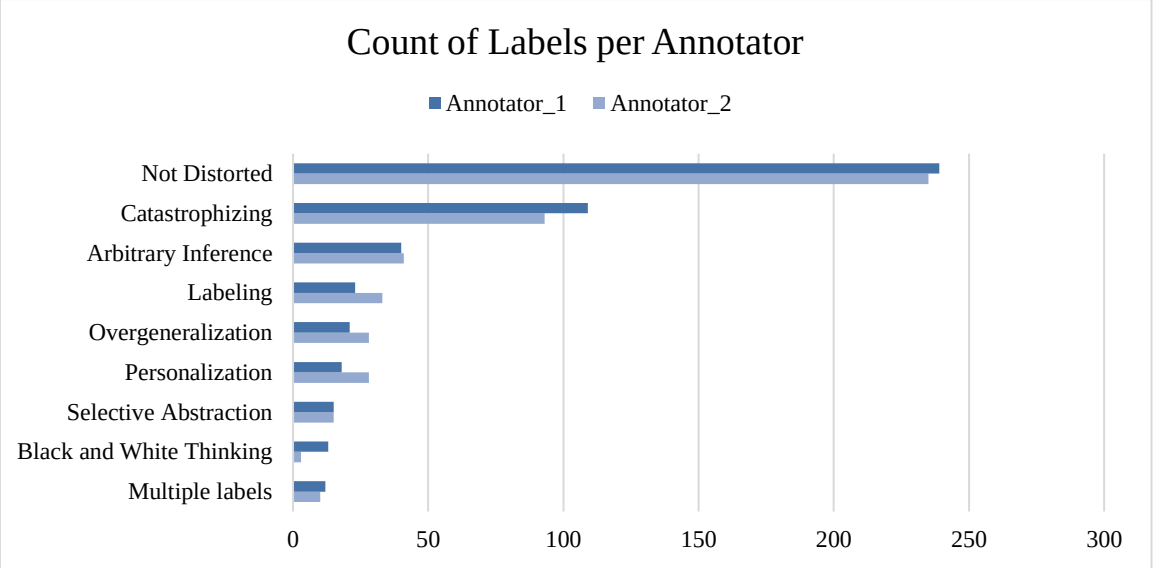


Figure 3. Distribution of labels per annotator in each label category.

As can be seen from Figure 3, most of the posts were annotated by both annotators into Not Distorted category, meaning that most of the posts did not convey any cognitive distortions. Nevertheless, the most prevalent distinct cognitive distortion noted by both annotators was Catastrophizing. For other distortions, the count remained below 50. There were also some posts that were given multiple labels by annotators but their proportion in the dataset was marginal. Also, there were only a few posts categorized into Black and White Thinking category.

4.2 Label Harmonization Techniques

The aim of the label harmonization is to get a set of label(s) per each text in the dataset in a harmonized way. In this work, three methods for harmonizing annotators' labels were used. These methods are defined as follows: the union method, the methodological intersection and the complete intersection. All these methods can be summarized by the Table 3.

Table 3. Label harmonization techniques example table.

Annotation_1	**Annotation_2**	**Union**	**Methodological intersection**	**Complete inter-section**
Not Distorted	Labeling	Labeling	Not Distorted	-
Not Distorted	Not Distorted	Not Distorted	Not Distorted	Not Distorted
Catastrophizing	Labeling	Catastrophizing, Labeling	Not Distorted	-
Catastrophizing, Labeling	Labeling	Catastrophizing, Labeling	Labeling	Labeling
Not Distorted	None	Not Distorted	Not Distorted	Not Distorted
Catastrophizing	None	Catastrophizing	Catastrophizing	Catastrophizing

As for the union harmonization method, the implemented idea is quite simple – the union of both annotator labels is taken. In case there was one label belonging to the Not Distorted category and another label was some distortion, existing distortion was chosen.

In case of the methodological intersection, the intersection of labels was taken. However, if the annotator labels did not match, it was assumed that there is no intersection and, therefore, the Not Distorted label was used. As of this method, the Not Distorted category is also chosen in case when one annotator labelled text as Not Distorted and another annotator as some distortion, as the presence of a distortion rules out the Not Distorted label. This method tends to be skewed towards the Not Distorted label which may not be the best option for continuing with classification.

The complete intersection technique is similar to the methodological intersection method in a way that both use the intersection of labels. However, according to the complete intersection technique, if the annotators did not agree, the result was empty and this text was disregarded. According to this method, if one label was None and another annotator had given some label to the text, the result should be empty according to this method. However, as the dataset is relatively small in size, in this case, the existing label was taken into use.

4.3 Preliminary Experiment Method Description

As mentioned earlier, two experiments were carried out in this work. Figure 4 explains the process flow for the preliminary experiment (seen also in bigger format under Appendix I).

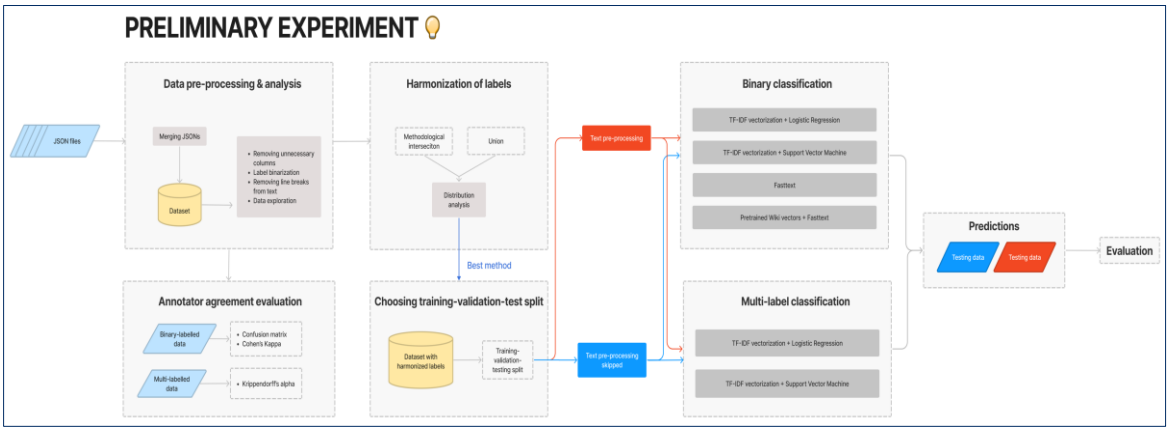


Figure 4. Preliminary experiment process schema.

In the preliminary experiment, firstly, data pre-processing and analysis was done. This included merging JSON files into a unified dataset, removing unnecessary columns, removing line breaks from the text and exploring the dataset. In order to perform classification, data needed to be in the format of 0-s and 1-s, therefore, label binarization was performed on the dataset as well. So, for each distortion category, a separate column was added where 1 denoted the existence of this distortion and 0 denoted the absence of this distortion per text. So, for example, if there was a text that belonged into two distortions, 1-s were marked in both distortion columns for this text, whereas other columns contained 0 values for that text. Also, an additional column was added to the dataset denoting whether text is distorted or not (binary form).

After that, inter-annotator agreement was evaluated in the binary and multi-label setting by using confusion matrix, calculating Cohen's Kappa and Krippendorff's Alpha. In order to

perform text classification, the annotator labels needed to be harmonized. In this experiment, two label harmonization techniques – the methodological intersection and union – were used. Best technique was chosen based on the distribution of labels.

The dataset was then divided into training, validation and test set in a stratified way. For this, *scikit-learn*'s function *train_test_split* with parameter *stratify* was used, where *stratify* was set equal to a vector of binarized labels. Splits in different sizes were experimented and one of them was chosen to further perform classification. In order to see, whether text pre-processing has any impact on classification results, the data was also pre-processed. This included following steps: lowercasing, removing extra whitespaces, tokenizing, lemmatizing and joining tokens together by whitespace. For both datasets – with and without pre-processing – binary and multi-label classification was performed. In binary and multi-label setting, logistic regression and support vector machine with TF-IDF vectorization and hyperparameter tuning was used. In addition, fasttext and fasttext with pre-trained Wiki vectors was used with manual hyperparameter tuning in binary setting. The idea of binary classification was to predict whether the text contained some distortion or not. The idea of multi-label classification was to predict different types of cognitive distortions from the text (excluding not distorted texts).

The best hyperparameters for logistic regression and SVM were found by using scikit-learn's *GridSearchCV* function with *scoring* parameter set to *'f1_weighted'*. In case of TF-IDF, *max_features* and *min_df* parameter were tuned for both classifiers. Parameter *max_features* considers top frequent words across the corpus and *min_df* parameter ignores terms that appear in less documents than the defined treshold[6]. In case of *max_features* parameter, a list of values [5000,4000,3000,2000,1000,500,5] were given to be tuned. For *min_df* parameter, a list of values [1,3,5,10] were given to be tuned. In case of logistic regression, parameter *C* (regularization strength) was also tuned, given a parameter list of [0.01,0.03,0.05,0.1,0.3,0.5,1,3,5,10,30,50]. Only parameter *solver* had a statical value of *liblinear* for logistic regression, since by documentation[7] it works best with small datasets. In case of SVM classifier, three parameters were tuned: regularization parameter *C*, kernel type parameter *kernel*, and kernel coefficient *gamma* in case of rbf kernel. The parameter values used for *C* were [0.1, 1, 10, 100, 1000], ['linear', 'rbf'] for *kernel* and [1000, 100, 10, 1, 0.1, 0.01, 0.001, 0.0001] for *gamma*. For both classifiers, *random_state* parameter was set to 1 for the results to be reproducible.

In case of fasttext, manual hyperparameter tuning was executed with simple for loops on the training and validation data. The parameters chosen to be optimized were the learning rate *lr* and the number of epochs *epoch*. In case of *lr*, a list of values [0.1,0.25,0.5,0.75,1.0] were given to be optimized. In case of *epoch*, a list of values [5,10,15,20,25,50] were given to be optimized. In order to see if using pre-trained vectors can improve classification results, fasttext with pre-trained vectors was also used for classification. The vectors were obtained from fastText's official documentation page[8]. In this work, one million word vectors trained on Wikipedia data with subword information was used. The vectors originate from Mikolov *et al.* (2017)'s work. In case of a fasttext model, parameter *dim* with value 300 and parameter *pretrainedVectors* with reference to the vector file were used. For fasttext with pre-trained vectors, the number of epochs and learning rate were also tuned similarly to the ordinary fasttext model. The best models were chosen by the highest F1-weighted score.

[6] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
[7] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[8] https://fasttext.cc/docs/en/english-vectors.html

In the case of multi-label classification, fasttext was not used. It was experimented in this work, however, it did not seem to support multi-label classification while making experiments. In the case of logistic regression and SVM, multi-label classification was done by using *scikit-learn*'s *MultiOutputClassifier* function. After finding the optimal parameters in binary and multi-label setting, predictions were made on the test set and evaluated. Results of the classifiers were then compared.

4.4 Primary Experiment Method Description

After performing the preliminary experiment, the primary experiment was also carried out. Figure 5 describes the process flow for this experiment (seen also in bigger format under Appendix II).

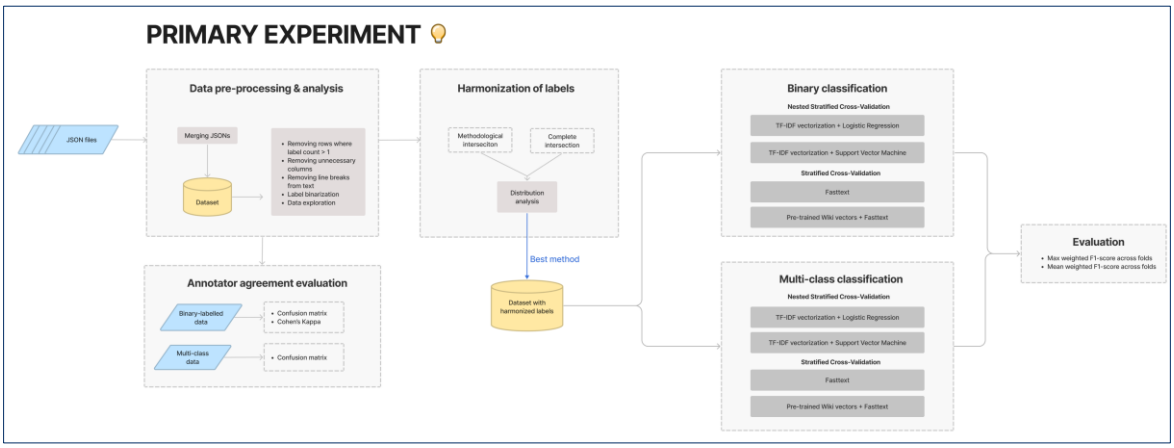


Figure 5. Primary experiment process schema.

In the primary experiment, firstly, data pre-processing and analysis was carried out. This included removing rows where the label count was bigger than one, in order to remove multi-labelled text from the dataset. As a result, out of 496 rows in the initial dataset, after removing rows with multiple labels, 473 rows remained in the dataset. The pre-processing and analysis also included same steps already mentioned under the preliminary experiment description. Then, inter-annotator agreement was evaluated. Similarly to the preliminary experiment, Cohen's Kappa and confusion matrix was calculated for binary-labelled data. However, in the primary experiment, confusion matrix was also found for the multi-class labelled data. After that, the harmonization of labels was done by using the methodological intersection and the complete intersection methods and the best method by analyzing the distribution of labels was chosen. This resulted in a dataset with harmonized labels. Then, classification in binary and multi-class setting was performed and the results were evaluated. In this experiment, text pre-processing was not used (reasons explained later in this work).

Contrary to the preliminary experiment where the dataset was divided into training, validation and test set, in this experiment, a different method was used. As the dataset used in this experiment after performing label harmonization resulted only in size of 216, it did not seem to be optimal to create such splits anymore as in the preliminary experiment. Instead, a method called nested stratified cross-validation was used. This method allows to use the whole dataset and find optimal parameters without creating the splits manually, which seems to be much optimal in this experiment.

According to this method, the dataset was divided into training and test splits by using *scikit-learn*'s function *StratifiedKfold*. Per each outer fold, an inner cross-validation was done using the same function. For the inner folds, function *GridSearchCV* was used for finding the optimal parameters (parameter *scoring* was set to *'f1_weighted'*). After the best parameters were found, the model was fitted to the whole training set and predictions were made on the outer test set. After this procedure, mean and maximum F1-weighted score were printed out. When it came to the fasttext models, this method was adapted. For fasttext, nested stratified cross-validation could not be used as it uses *GridSearchCV*. Instead, stratified cross-validation was made without using *GridSearchCV*. For the sake of simplicity, for fasttext, the optimal parameters found in the preliminary experiment were used.

In this experiment, classification in binary and multi-class setting was performed. In binary setting two classifcation tasks were made and in multi-class setting some labels were grouped together (see in detail from the results section). Logistic regression, SVM, fasttext and fasttext with the pre-trained Wiki vectors were used in both settings. Cross-validation results were evaluated by models' weighted F1-scores.

5. Results

In this chapter, results of the preliminary and the primary experiment are being discussed. The code of the experiments can be accessed from Linda Katariina Grents' GitHub repository[9].

5.1 Preliminary Experiment

In the preliminary experiment, the whole dataset including 496 rows was used.

5.1.1 Annotator Agreement and Harmonization of Labels

Firstly, confusion matrix was found from the binarized data (Distorted vs Not Distorted texts) to assess how well the annotators have agreed with each other. It is seen from the confusion matrix (Figure 6) that mostly annotators were labeling texts into some distortion category and a little bit less into the Not Distorted category. However, a lot of confusion remained between the annotators.

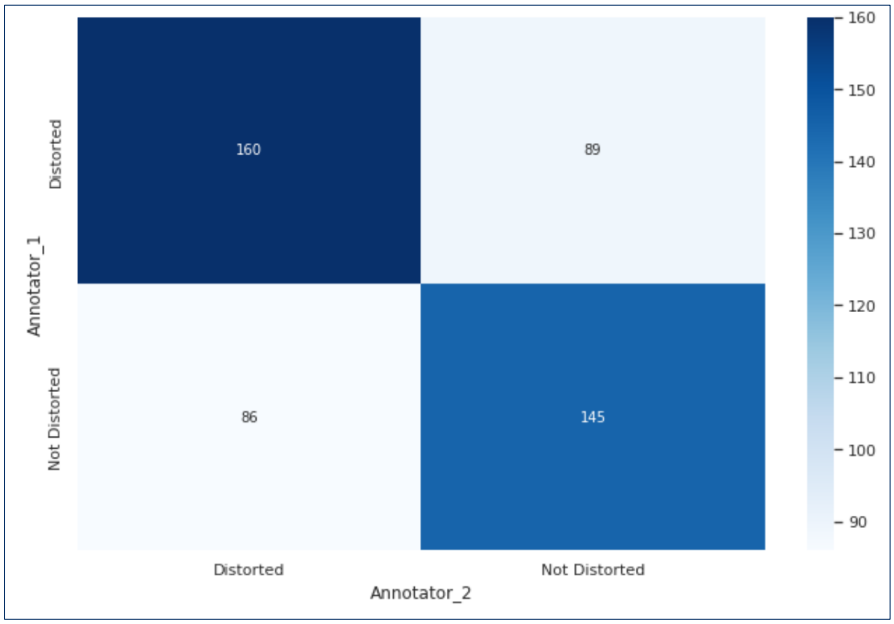


Figure 6. Binary confusion matrix between annotator_1 and annotator_2.

In order to further investigate the degree of agreement between the annotators and whether it happened by chance or not, Cohen's Kappa and Krippendorff's Alpha were calculated. Cohen's Kappa was calculated in a binary setting (excluding rows with None) which resulted in a value of 0.27, which, according to Landis and Koch (1977), can be interpreted as fair agreement. Then, Krippendorff's Alpha was found using masi distance. Contrary to Cohen's Kappa, Krippendorff's Alpha can handle missing data. The metric resulted in score of 0.18 which is quite low and does not indicate very good reliability of the data.

In order to proceed with the classification task, label harmonization needed to be done. In this experiment, the methodological intersection and union were performed and compared. The results of label harmonization for this experiment can be seen from Table 4.

[9] https://github.com/lindakatariina/master_thesis

Table 4. Label distribution as per the methodological intersection and the union method.

Label	Methodological intersection	Union
Not Distorted	421	154
Catastrophizing	47	168
Labeling	13	56
Overgeneralization	4	52
Personalization	4	43
Arbitrary Inference	3	88
Selective Abstraction	2	33
Black and White Thinking	2	21
Total	**496**	**615**

It is seen from the table that the methodological dataset is highly skewed towards Not Distorted label. Also, Selective Abstraction and Black and White Thinking distortions appear in the dataset only twice, so in case the dataset was split into training, validation and test set, one label would be missing from one of the sets. Also, there are quite few distinct labels in general. The union dataset, however, contains much more distortions. The union labels in the table represent count of each label in the dataset. So the sum of these counts do not add up to the number of texts in the datasets. Minimum of occurrance of a label according to the union method is 21 which is much more than was achieved by the methodological intersection. As the distribution of labels is more even for the union dataset and there are more labels available, the union dataset was chosen to use in further classification tasks.

5.1.2 Binary Classification

The idea of binary classification in this work was to predict the existence of some distortion from the text (categorize text into Distorted or Not Distorted category). The support values for labels across training, validation and test set can be seen from Table 5.

Table 5. The support values of Not Distorted and Distorted labels across chosen split in the binary setting.

Label	Training set	Validation set	Testing set
Distorted	273	34	35
Not Distorted	123	16	15
Total	**396**	**50**	**50**

The split was chosen as 80% training, 10% validation and 10% testing. Classification results from both – on pre-processed and original texts – on the test set can be found from Table 6.

Table 6. Binary classification weighted F1-scores on pre-processed and original dataset (predictions made on test set).

Model	Model's weighted F1-score on the original text	Model's weighted F1-score on the pre-processed text
Logistic Regression	0.67	0.70
Support Vector Machine	0.69	0.69
Fasttext (max F1 from 10 for loop)	0.61	0.64
Fasttext with pre-trained vectors (max F1 from 10 for loop)	0.58	-

In case of logistic regression, best hyperparameter set turned out to be *C*=10, *max_features*=5000, *min_df*=10. With this parameter set, the classifier reached a weighted F1-score of 0.67 on test set. Precision and recall of the Not Distorted category turned out to be 0.46 and 0.40 respectively and for the Distorted category 0.76 and 0.80 respectively. In case of logistic regression on pre-processed text, the F1-weighted score value increased by 0.03 which was less than expected (precision and recall also rose slightly but not significantly). In case of SVM, the best hyperparameter set turned out to be *C*=10, *kernel*=rbf, *gamma*=0.1, *max_features*=500, *min_df*=5. Using this set of parameters, the weighted F1-score of this classifier reached a score of 0.69 on the test set which is slightly (0.02) more than for logistic regression. Precision and recall of the Not Distorted category turned out to be 0.50 and 0.40 respectively and for the Distorted category 0.76 and 0.83 respectively. Training this classifier on the pre-processed dataset did not have any effect on the weighted F1-score, as it remained the same.

Fasttext hyperparameters, as mentioned, were optimized manually on the validation set. Per each parameter set, an inner 10-run for loop was also executed. The reason being that for fasttext, there is no option to set a parameter similar to *random_state* which would make results reproducible (as was used for logistic regression and SVM). So, for each run, the results would be different. Thus, for each 10-run for loop, an average weighted F1-score was saved into a list with respective parameter set. After each parameter set was run, the best average weighted F1-score was printed out with the respective parameter set on the

validation data. The best parameter set was taken and predictions were made on the test set again with a 10-run for loop with these parameters. A maximum and average weighted F1-score was taken from these results. According to the maximum F1-score, fasttext performed a bit worse than logistic regression and SVM, as the maximum F1-score resulted in a value of 0.61. Training the same classifier on the pre-processed text increased the maximum F1-weighted score only marginally (by 0.03). Surprisingly, fasttext with pre-trained vectors performed the worse on the test set, as its weighted F1-score resulted only in value of 0.58. Fasttext with pre-trained vectors was not trained on the pre-processed dataset.

Models' training times were compared as well. Finding the best hyperparameters and training for logistic regression on the original dataset took around 80 seconds and on the pre-processed dataset approximately 104 seconds. For SVM, however, choosing optimal parameters took much longer. For SVM on the original dataset, training and finding optimal parameters took 1539 seconds and 1525 seconds on the pre-processed dataset, which is approximately 25 minutes. Fasttext took 65 seconds on both datasets. Fasttext with pre-trained vectors took 1072 seconds which is almost 18 minutes.

5.1.3 Multi-label Classification

As the algorithms could predict the existence of some distortion from the text to some extent, it was decided that it would be interesting to see, whether these algorithms could detect different types of distortions from the text as well. Thus, the aim of multi-label classification in this work was to predict 7 different types of cognitive distortions from the text. The support values for labels across training, validation and test set can be seen from Table 7.

Table 7. The support values of distortions across training, validation and test set in multi-label setting.

Distortion	Training set	Validation set	Testing set
Black and White Thinking	16	2	3
Selective Abstraction	24	3	6
Personalization	35	5	3
Labeling	42	8	6
Overgeneralization	44	4	4
Arbitrary Inference	68	10	10
Catastrophizing	138	12	18
Total	**367**	**44**	**50**

The results of multi-label classification with logistic regression and SVM can be seen from Table 8.

Table 8. Multi-label models' F1-scores per distortion and models' overall weighted F1-scores on the pre-processed and original dataset (predictions made on the test set).

Distortion	F1-score on the original text		F1-score on the pre-processed text	
	Logistic Regression	Support Vector Machine	Logistic Regression	Support Vector Machine
Arbitrary Inference	0.21	0.33	0.22	0.36
Black and White Thinking	0.00	0.00	0.00	0.00
Catastrophizing	0.21	0.21	0.33	0.30
Labeling	0.25	0.20	0.25	0.22
Overgeneralization	0.00	0.29	0.00	0.00
Personalization	0.00	0.50	0.00	0.33
Selective Abstraction	0.00	0.00	0.00	0.00
Model's weighted F1-score	**0.15**	**0.22**	**0.19**	**0.23**

In the multi-label classification task, hyperparameter tuning was also done, using the same hyperparameter sets as in the binary classification task. As for logistic regression, the following set of hyperparameters was found to be the most optimal on the validation set: *C*=50, *max_features*=500, *min_df*=10. The weighted F1-score of the model on the test set turned out to be very low (0.15) with these parameters. When observing the per distortion F1-scores, these turned out to be low as well. The highest per-distortion F1-score was achieved for Labeling distortion (0.25), followed by Catastrophizing (0.21) and Arbitrary Inference (0.21). Labeling distortion had the highest precision rate amongst all distortions (0.50). The model could not detect any other distortion from the text. As for the classification on the pre-processed text, the optimal parameters remained the same, except in this case, the optimal *min_df* turned out to be 3. On the pre-processed dataset, the model's overall weighted F1-score increased by 0.04 with this classifier, which is slightly more than on the original dataset. Arbitrary Inference's F1-score increased by 0.01 and Catastrophizing distortion's F1-score increased by 0.12. Per distortion precision rates remained in the same range (between 0.25-0.50). Both models could predict only similar distortions. The highest precision rate turned out to be 0.50 in both cases for the Labeling distortion.

SVM turned out to perform slightly better than logistic regression in this task. On the original dataset, the model's overall F1-weighted score turned out to be 0.22 which is still low but slightly better than the logistic regression's score. The optimal parameters in this case turned out to be the following on the validation set: *C*=10, *kernel*=linear, *max_features*=1000, *min_df*=10. SVM could detect two distortions more than logistic regression. Distortions that were detected by SVM are the following (by descending order of F1-scores): Personalization (0.50), Arbitrary Inference (0.33), Overgeneralization (0.29), Catastrophizing (0.21) and Labeling (0.20). The per-distortion precision rates remained between 0.25-0.40. It is surprising that Personalization distortion reached the highest F1-score as there were only three examples of this distortion in the test set. On the pre-processed dataset, the overall model's weighted F1-score improved only by 0.01. This model, however could not detect Overgeneralization distortion anymore. Other distortions' F1-scores

increased slightly, except for Personalization, for which the F1-score decreased. The precision rates for Arbitrary Inference, Catastrophizing, Labeling and Personalzation were all 0.33. The recall values for before-mentioned distortions remained between 0.17-0.40.

Similarly to the binary classification, finding optimal parameters and training logistic regression took far less time than SVM. For logistic regression, hyperparameter tuning and training took 62 seconds on the original and 36 seconds on the pre-processed dataset, whereas SVM took 1577 seconds on the original and 1598 seconds on the pre-processed dataset, which is approximately 26 minutes.

5.2 Primary Experiment

For before-mentioned reasons, in this experiment, all rows where there was more than one label per text were removed from the dataset in the primary experiment, which resulted the dataset in size of 473 rows. In this experiment, pre-processing (lowercasing, removing extra whitespaces, tokenizing, lemmatizing and joining tokens together by whitespace) was not carried out anymore. It was decided as such because the classification results on the pre-processed dataset did not ameliorate significantly in the preliminary experiment compared to on the original dataset. Also, the number of models would have become too numerous to compare and analyze.

5.2.1 Annotator Agreement and Harmonization of Labels

Firstly, binary confusion matrix between annotator_1 and annotator_2 was calculated. This confusion matrix displays how well the annotators agreed upon labeling texts into Distorted or Not Distorted category. The results can be seen from Figure 7. It is seen from the figure that annotators mostly agree upon whether the text belongs to category Not Distorted, followed by agreeing upon the Distorted category. Even though the annotators agree on a lot of texts, similarly as was seen in the preliminary experiment, it is also seen that they disagree a lot as well.

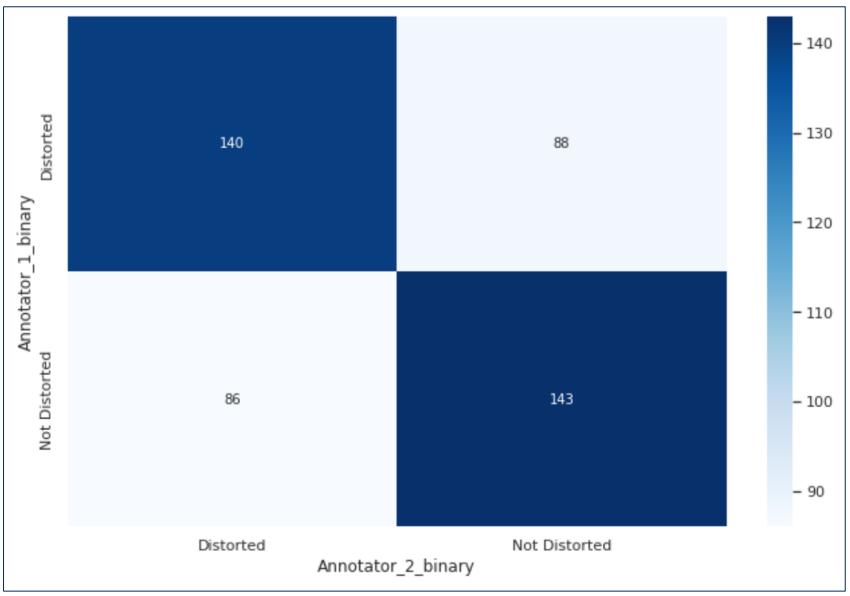


Figure 7. Binary confusion matrix between annotator_1 and annotator_2.

For further investigation, confusion matrix between different distortions was also calculated (excluding the Not Distorted class). The confusion matrix can be seen from Figure 8. It is seen from the figure that the annotators mostly agree upon the Catastrophizing distortion. The annotators also agree upon the Labeling distortion, however, not as much as upon Catastrophizing. In addition, it is seen that the annotators mostly are confused between Catastrophizing and Arbitrary Inference distortion. They also confuse the Catastrophizing distortion with Overgeneralization.

Figure 8. Multi-class confusion matrix between annotator_1 and annotator_2.

To further estimate the annotator agreement, Cohen's Kappa was calculated in the binary setting (between categories Distorted vs Not Distorted). Rows which had missing data were ignored by the calculation. Cohen's Kappa resulted in value 0.24, which, according to Landis and Koch (1977), can be interpreted as fair agreement which is similar result obtained in the preliminary experiment, only slightly less.

To use the data for classification, the annotator labels needed to be harmonized. In this experiment, the methodological intersection and the complete intersection methods were used. The results of the harmonization can be seen from Table 9. It is seen from the table that distortions have the same count according to both methods. However, the methodological intersection method preserves more Not Distorted texts than the complete intersection method (409 vs 152). As the complete intersection is mostly used in related woks and the distribution between Not Distorted texts and Distorted labels is more even, the complete intersection results were decided to be used in further classification tasks.

Table 9. Count of labels per methodological and complete intersection method.

Label	Methodological intersection	Complete intersection
Not Distorted	409	152
Catastrophizing	43	43
Labeling	9	9
Overgeneralization	4	4
Personalization	4	4
Selective Abstraction	2	2
Arbitrary Inference	1	1
Black and White Thinking	1	1
Total	**473**	**216**

It is assumed that as per the complete intersection there is 100% agreement between annotator_1 and annotator_2, since all labels that the annotators disagreed upon were thrown out from the dataset, binary classification (predicting distorted text) should give better results than results obtained in the preliminary experiment. This assumption can be further validated in following sub-paragraphs where classification on the complete intersection data will be executed. However, the results of the preliminary and primary experiments cannot be directly compared, as they use slightly different datasets.

5.2.2 Binary Classification

As for the binary classification, it was decided that it would be useful to create two classification tasks. The first classification task would be to predict the existence of some distortion in the text, similarly as was done in the preliminary experiment, thus, predicting whether a text belongs to Distorted or Not Distorted category. In addition, as there were so few distinct distortions left after creating the complete intersection of annotator labels and the most prevalent distortion in our dataset was Catastrophizing, it was thought to be useful to also predict whether a text belongs to either Catastrophizing or Other distortion (all other distortions grouped together). Therefore, in the second binary classification task, all other distortions besides Catastrophizing were grouped into the Other class and classification was made between the Other distortions and Catastrophizing distortion. The final results of the binary classification can be found from Table 10.

Table 10. Results of binary classification in the primary experiment.

Classification task	Logistic Regression		Support Vector Machine		Fasttext		Fasttext with pre-trained vectors	
	F1 max	**F1 mean**	**F1 max**	**F1 mean**	**F1 max**	**F1 mean**	**F1 max**	**F1 mean**
Distorted vs Not Distorted	0.81	0.66	0.84	0.66	0.77	0.64	0.85	0.68
Catastrophizing vs Other	0.84	0.58	1.0	0.64	0.59	0.54	0.81	0.65

As for predicting the texts into Distorted and Not Distorted categories, the results of the classifiers were quite similar. Per each classifier, 10 outer and 3 inner splits were used, except for Fasttext where inner splits were not used. The highest F1-scores across classifiers ranged from 0.64 up to 0.85 and the mean F1-scores ranged between 0.66-0.68. In this classification task, fasttext with pre-trained vectors performed the best as its highest F1-score across folds reached 0.85 and mean score reached 0.68. The results of Distorted vs Not Distorted prediction seem somewhat surprising, as the results did not seem to ameliorate that much compared to the preliminary experiment, even though texts where the annotators disagreed were thrown out from the dataset. Nevertheless, it needs to be noted that in this task there was an imbalance towards the Not Distorted category in the folds. Across the folds, the classifier would mostly predict Not Distoted category, as its precision, recall and F1-score were almost always higher. For logistic regression, optimal parameter *C* value was usually 3 or 5 across folds. For SVM, optimal values for parameter *C* were usually 10, 100 or 1000; optimal value for parameter *kernel* was rbf and optimal value for *gamma* were 0.001, 0.01, 0.1 or 1. For simple Fasttext model, learning rate of 0.25 and epoch of value 25 was used. For Fasttext with pre-trained vectors, learning rate of 1.0 and epoch of value 10 was used with *dim* value of 300.

In predicting texts into Catastrophizing vs Other distortions, the results seem to be more varied. Similarly to previous task, per each classifier, 10 outer and 3 inner splits were used, except for fasttext where inner splits were not used. The highest F1-scores across classifiers ranged from 0.59 up to 0.84 and the mean F1-scores ranged from 0.54-0.65. In this classification task, the highest F1-score was reached by using SVM (1.0) and the best mean F1-score (0.65) was reached by using Fasttext with pre-trained vectors. There were always more Catastrophizing labels than Other distortions (by looking at the support values) which also have an effect on the classification results. The classifiers could mostly predict the Catastrophizing label and sometimes Other. For logistic regression, optimal parameter *C* value was usually 3 or 5 across folds. For SVM, optimal values for parameter *C* was usually 10, 100 or 1000; optimal value for parameter *kernel* was either linear or rbf and optimal value for *gamma* was 0.01, 0.1 or 1. For simple fasttext model, learning rate of 0.25 and epoch of value 25 was used. For fasttext with pre-trained vectors, learning rate of 1.0 and epoch of value 10 was used with *dim* value of 300.

5.2.3 Multi-class Classification

As the count of some distortions was very low in the dataset after performing the complete intersection, soon it became clear that it is not reasonable to carry out multi-class classification task where distinct types of cognitive distortions are to be predicted. There were some distortions in the dataset that were only present in the dataset once. Thus, it was

decided that it would be more reasonable to group some distortions together, as was done with Catastrophizing vs Other prediction in the previous section, and create multi-class classification based on that.

Therefore, two multi-class classification tasks were made. So, the aim was to predict the most prevalent distortions and group other distortions into one category – Other. To be more precise, the aim of this task was to predict Catastrophizing, Labeling and Other distortions. The second classification task was to predict Catastrophizing, Other (all distortions besides Catastrophizing grouped into one) and Not Distorted texts. The results of these classification task results can be found from Table 11.

Table 11. Multi-class classification results in the primary experiment.

	Logistic Regression		**Support Vector Machine**		**Fasttext**		**Fasttext with pre-trained vectors**	
Classification task	**F1 max**	**F1 mean**	**F1 max**	**F1 mean**	**F1 max**	**F1 mean**	**F1 max**	**F1 mean**
Catastrophizing vs Labeling vs Other	0.69	0.57	0.79	0.60	0.79	0.59	0.79	0.59
Catastrophizing vs Other vs Not Distorted	0.77	0.59	0.77	0.62	0.74	0.61	0.73	0.62

In the first classification task where the Catastrophizing vs Labeling vs Other distortion prediction was made, the label distribution was 43, 12 and 9 respectively. In this task, the number of outer splits was set to 9 as the Labeling distortion was occurring only 9 times in the dataset. The inner split was set to 3. For fasttext models, no inner splits were used. It is seen from the classification results that the highest F1-scores ranged between 0.69 and 0.79, whereas the mean F1-scores ranged between 0.57-0.60 across classifiers. It is of worth noting that the classifiers could almost never predict Labeling or Other category. It is an expected result to some extent as there were simply more texts belonging to Catastrophizing category than to Labeling or Other category. The highest weighted F1-score was achieved with fasttext and fasttext with pre-trained vectors. Logistic regression and SVM best hyperparameters remained more or less in the same range as in the binary classification tasks.

In the second classification task where the Catastrophizing vs Other vs Not Distorted prediction was made, the label distribution was 152, 43 and 21 respectively. In this task, the number of outer splits was set to 10 and inner set to 3, except for fasttext models where no inner splits were defined. It is seen from the results that the highest F1-scores across classifiers ranged between 0.73-0.77, whereas the mean F1-scores across classifiers ranged between 0.59-0.62. The highest F1-score was achieved with logistic regression and SVM reaching score of 0.77, whereas the highest mean score of 0.62 was achieved by SVM and fasttext with pre-trained vectors. It is of worth noting that logistic regression could mostly predict only texts in the Not Distorted category and SVM could sometimes also predict the Catastrophizing label. Simple Fasttext model, similarly to SVM, could mostly predict the Catastrophizing label and Not Distorted texts. Fasttext with pre-trained vectors could predict sometimes the Catastrophizing distortion and Not Distroted texts and sometimes Not

Distorted and texts belonging to the Other category. In conclusion, the results seemed to vary across classifiers, however, mostly, the classifiers could not predict the Other category and mostly could predict the Not Distorted category. These results do not seem to be surprising, as the classifiers had more data to learn texts belonging to the Not Distorted category, as the support was simply the highest for this label. Logistic regression and SVM best hyperparameters remained more or less in the same range as in the binary classification tasks.

6. Discussion of Results

According to the preliminary and primary experiment, it can be concluded that the annotators in this work did not reach a good level of agreement in neither experiments. The annotators did not agree upon 36% of the posts in the preliminary experiment and upon 38% according in the primary experiment (before label harmonization). Taking into account both experiments, Cohen's Kappa did not reach over 0.27 and Krippendorff's Alpha was low (0.18), indicating that the data is not of high reliability and agreement happened slightly over chance. Also, by looking at confusion matrix in the primary experiment, it was seen that, mostly, the annotators could only unanimously annotate the Catastrophizing distortion. There was much confusion between annotators when it came down to other distortions, indicating that there could have been issues with understanding the annotation assignment.

Shreevastava & Foltz (2021) also did not achieve very good inter-annotator agreement score. Sochynskyi (2021), however, reached much better Cohen's Kappa score (0.569) on a similar dataset. It is of worth noting that Sochynskyi worked out the annotation guideline himself. Therefore, he might have been more systematic in the annotation process. The supervisor of this thesis, Kairit Sirts (PhD), was another annotator in his work. Also, Sochynskyi decided not to annotate very long texts – only posts ranging between 200 up to 1500 characters were annotated. However, the dataset used in present work was retrieved from Reddit randomly and no such processes were made as in Sochynskyi's. In addition, the dataset used in the present work was annotated by students who, most probably, do not have any background in clinical psychology. Furthermore, the students may not be the best possible annotators available, since they might lack motivation to thoroughly assess the texts. It might be assumed that some of the students simply wanted to pass the homework assignment. Therefore, there is a possibility that their annotations simply cannot be reliable. It is assumed that using professional annotators could raise the quality and reliability of the ground truth labels. This assumption can be supported by the work of Ahjaj *et al*. (2022) where professional psychotherapists annotated the data and the Cohen's Kappa resulted in score of 0.817, which is much higher than achieved in this work.

Interestingly, having a dataset with annotations of good quality is a challenge on its own in the field of natural language processing. This problem brings on much debate in the scientific literature. For example, Dumitrache *et al*. (2015) see this issue especially in the medical field where expert level assessment is usually expensive. They propose a method – CrowdTruth – as a way to get annotations in a cheaper and faster way than using medical experts. According to this method, the ground truth labels are gathered with crowd-sourcing via CrowdFlower platform. They show that one sentence needs to be annotated by at least ten workers to achieve the highest annotation quality, which is more than using 2-5 medical experts per task but which is, however, cheaper. They claim that using more annotators can capture ambiguity well. So, it might be worth in the future to try to crowdsource the ground truth labels via crowd-sourcing to capture annotation ambiguity.

Another reason for low inter-annotator agreement score could be that the annotation guideline may have not been that clear to the annotators. This can be ameliorated by asking feedback from the students how well they understood their assignment and make corrections to the guideline based on that. This can be seen from the fact that annotation guideline stated that each text can have one possible label but the students annotated some texts into multiple distortion categories in some cases. In the future, the annotation system should make constraints in such cases, allowing the annotator to only insert one label if the task states so.

The inter-annotator agreement score may also be low as this annotation task itself is difficult in nature. It can be assumed that the annotation task gets more complex as the number of

distortions to identify grows. In this work, the annotators had to choose between seven distortions and also identify the texts with no distortions in it. It may be that with lower number of distortion categories, it might have been an easier task for the annotators. For example, by including only the most prevalent distortions found in the literature, it might have been easier to carry out the annotation procedure. In addition, sometimes, the texts can be categorized into multiple distortions which also makes the task more difficult.

From the results of this work it can be seen that predicting the existence of some distortion in the text (Distorted vs Not Distorted prediction) was more or less successful. It was seen from the primary experiment that fasttext with pre-trained vectors produced the highest F1-scores (max 0.85, mean 0.68). This is somewhat comparable with Sochynskyi's (2021) results where a score of 0.71 was reached by using simple fasttext model. It is of worth noting that there was significantly more data available for training and predicting in Sochynskyi's work (1931 datapoints) than in this work where only 216 datapoints were available to use (in the primary experiment). Since the datasets are a bit similar in essence, it might be assumed that with more data at hand, fasttext with pre-trained vectors could produce even better results.

On the other hand, predicting distinct cognitive distortions from text was rather unsuccessful. Even though the preliminary experiment results cannot be taken as reliable results as in the primary experiment, it was seen from the preliminary experiment that classifiers performed poorly in the multi-label classification task and could only identify some distortions from the text. The models' overall F1-scores could not even reach above 0.23. It can be concluded that if annotating texts into distinct distortion categories was confusing for humans, the models simply cannot produce any better results. It was also seen from related works (Sochynskyi, 2021; Shreevastava & Foltz, 2021) that predicting distinct distortions was rather unsucessful. It might be concluded that this task is simply very difficult for humans and algorithms.

It was seen that even in the primary experiment, where only unanimously annotated data was used, the multi-class prediction did not perform the best. The results seemed to be quite average. By predicting Catastrophizing, Labeling and Other distortions, the classifiers could almost always predict only Catastrophizing label. Predicting texts as Catastrophizing, Other or Not Distorted produced also quite average results by looking at the models' maximum and mean F1-scores. In addition, in that task, classifiers could almost never predict all labels at once. This was probably due to low amount of available data and class imbalance. Considering these results, it might be concluded that it is not that reasonable to predict different types of cognitive distortions but rather predict the existence of some distortion in the text or some distortions that are most prevalent. In case there is a need to predict distinct types of distortions, definitely more data needs to be used so the classifiers could learn more. Classifiers simply did not have enough data to learn from in this work.

When it comes to the label harmonization, it seems that the complete intersection method could provide the best results. It was mainly also used in related works (Ahjaj *et al*., 2022; Aureus *et. al* 2021) as well. However, to use this method, one needs to have a large dataset at hand, since examples where annotators have disagreed upon will be eliminated from the dataset. It is also important that the annotations would be of high quality. The more the annotators agree upon labeling texts, the more data will remain at hand and the better results one might get upon classification.

The major constraint in this work was the dataset itself. The dataset used in this thesis is relatively small in size. It is also noisy as it originates from Reddit where people do not tend to write in correct English and, usually, add emoticons, links and other noise to the text.

There is not much work performed on such noisy datasets that is of knowledge to the author of this thesis. It is assumed that making cognitive distortion prediction from noisy and small datasets remains a challenge to be tackled. It may be that performing more pre-processing might raise the classification accuracy. However, it must be done in a well thought-through manner by not eliminating any important information from the original text. In this work, however, the pre-processing that was done did not provide significantly better results than on the original text.

7. Summary

The aim of this thesis was to predict cognitive distortions from posts derived from the Anxiety sub-reddit. In this work three research questions were raised:

1) How well have the annotators agreed on labeling cognitive distortions?
2) How well do supervised machine learning techniques predict cognitive distortions from a dataset that is relatively small in size?
3) How do this work results compare to previous works?

The original dataset used in this thesis resulted in 496 rows of data. The data was labelled by the students of University of Tartu in 2020. This work was divided into the preliminary and primary experiment. Both experiments covered inter-annotator agreement assessment. In the preliminary work, binary and multi-label classification was performed. In the primary experiment, multi-labelled data was thrown out from the dataset (which reduced dataset size) and binary and multi-class classification was done. Both experiments used different label harmonization techniques.

The results show that in this thesis annotators did not reach a good level of agreement by evaluating inter-annotator agreement (Cohen's Kappa resulted only in score of 0.24 in the primary experiment). Predicting the existence of cognitive distortions from the text in binary setting turned out to be, however, more or less successful, as fasttext with pre-trained word vectors as the best classifier amongst others could produce quite satisfactory weighted F1-scores (max 0.85, mean 0.68). However, predicting distinct cognitive distortions from the text remained a difficult task and did not produce very satisfactory results. The results were comparable to related works to some extent, as previous works also did not provide very good results when classifying text into distinct distortion categories either. However, the dataset used in this work was very limited in size and noisy which definitely had a negative effect on the classification performance. In order to predict distinct cognitive distortions, more data needs to be given to the classifiers. The quality and the size of the dataset was a major constraint in this work.

There are some suggestions for the future works as well. It is assumed that transfer learning techniques, for example BERT, could provide better results than supervised machine learning techniques on similar datasets. BERT could potentially give better results, as was seen from the work of Ahjaj *et al*. (2022) on arabic Twitter dataset. It would be interesting to see how its results compare for example with fasttext results. There is also a suggestion to think about whether it is actually necessary to predict different cognitive distortions from the text, as this and previous works do not tend to provide good results. It may be of more use to detect the existence of cognitive distortions to provide help to people who tend to have depression or anxiety issues more quickly and effectively. It might also be useful to only predict the most prevalent distortions from the text, as some distortions simply might not be that prevalent amongst people.

The biggest obstacle in this research area remains the lack of availability of a high-quality dataset. It is suggested in the future works to use professional annotators or use crowd-sourcing for labeling texts. More effort needs to be put into obtaining the best available dataset, since the quality of the data that is used for classification plays a crucial part in the classification results.

8. References

Alhaj, F., Al-Haj, A., Sharieh, A. & Jabri R. (2022). Improving Arabic Cognitive Distortion Classification in Twitter using BERTopic. *International Journal of Advanced Computer Science and Applications (IJACSA), 13*(1).

Aureus, J. P., Estuar, M. R. J. E., Mapua, D. C., Abao, R. P. & Cataluña, A. A. M. (2021). Determining Linguistic Markers in Cognitive Distortions from COVID-19 Pandemic-Related Reddit Texts. *2021 1st International Conference in Information and Computing Research (iCORE)*, 56-61.

B. Shickel, S. Siegel, M. Heesacker, S. Benton and P. Rashidi. (2020). Automatic Detection and Classification of Cognitive Distortions in Mental Health Text. *2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)*, 275-280.

Beck, A. T. (1963). Thinking and Depression: I. Idiosyncratic Content and Cognitive Distortions. *Archives of General Psychiatry, 9*(4), 324–333.

Beck, A. T., & Weishaar, M. (1989). Cognitive Therapy. In: Freeman, A., Simon, K.M., Beutler, L.E., Arkowitz, H. (eds). *Comprehensive Handbook of Cognitive Therapy*. New York: Springer.

Beck, J. S. (2011). *Cognitive Behavioural Therapy. Basics and Beyond*. Second Edition. New York: The Guilford Press.

Burns, D. D., M.D. (2012). *Feeling Good: The New Mood Therapy*. Harper Collins. https://www.scribd.com/book/163608148/Feeling-Good-The-New-Mood-Therapy?fbclid=IwAR2eVRc2Gwu5mhIYZ9BYnTSG4XgPOiggSE0pzUwLffCAUGr3ssJIPGWLCMg (15.05.2022)

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement, 20*(1), 37-46.

Cully, J. A., & Teten, A. L. (2008). *A Therapist's Guide to Brief Cognitive Behavioral Therapy*. Houston: Department of Veterans Affairs South Central MIRECC.

Dumitrache, A., Aroyo, L., & Welty, C. (2015). Achieving Expert-Level Annotation Quality with CrowdTruth: The Case of Medical Relation Extraction. [PDF] Achieving Expert-Level Annotation Quality with CrowdTruth: The Case of Medical Relation Extraction | Semantic Scholar (01.05.2022)

Gilchrist, A. (2017). *Machine Learning: Adaptive Behaviour Through Experience: Thinking Machines.* Book III.

Grave, E. (2016). *Releasing Fasttext*. Official Fasstext Blog site. Releasing fastText · fastText (01.05.2022)

Gulli, A., Kapoor, A., Pal, S. (2019). *Deep Learning with TensorFlow 2 and Keras - Second Edition: Regression, ConvNets, GANs, RNNs, NLP, and more with TensorFlow 2 and the Keras API.* 2nd Edition. Packt Publishing.

Hilbe, J. M. (2015). *Practical Guide To Logistic Regression*. New York: A Chapman & Hall Book/CRC Press.

Jager-Hyman, S., Cunningham, A., Wenzel, A., Mattei, S., Brown, G. K. & Beck, A. T. (2014). *Cognitive Distortions and Suicide Attempts*. Cogn Ther Res 38, 369–374. https://link.springer.com/article/10.1007/s10608-014-9613-0#citeas (30.04.2022)

Jayaswal, V. (2020). *Text Vectorization: Term Frequency — Inverse Document Frequency (TFIDF)*. Text Vectorization: Term Frequency — Inverse Document Frequency (TFIDF) | by Vaibhav Jayaswal | Towards Data Science) (01.05.2022)

Joshi, G., Singh, S., Varshney, P. & Pant, A. (2021). A Burgeoning Social Media Infodemic Amid COVID-19 Pandemic: A Cognitive Behavioural Perspective. *Indian Journal of Behavioral Sciences*. https://www.researchgate.net/profile/Gunjan-Joshi-9/publication/355932311_A_Burgeoning_Social_Media_Infodemic_Amid_COVID-19_Pandemic_A_Cognitive_Behavioural_Perspective/links/61852475a767a03c14f8b88c/A-Burgeoning-Social-Media-Infodemic-Amid-COVID-19-Pandemic-A-Cognitive-Behavioural-Perspective.pdf (01.05.2022)

Joulin, A., Grave, E., Bojanowski, P. & Mikolov, T. (2016). Bag of Tricks for Efficient Text Classification. 1607.01759.pdf (arxiv.org) (01.05.2022)

Jurafsky, D. & Martin, J. H. (2021). Speech and Language Processing. 3rd ed. draft. https://web.stanford.edu/~jurafsky/slp3/ (15.05.2022)

Krippendorff, K. (2004). Reliability in Content Analysis: Some Common Misconceptions and Recommendations. *Human Communication Research, 30*, 411-433.

Krippendorff, K. (2011). Computing Krippendorff's Alpha-Reliability. "Computing Krippendorff's Alpha-Reliability" by Klaus Krippendorff (upenn.edu) (01.05.2022)

Kulkarni, A., Chong, D. & Batarseh, F. A. (2020). 5 - Foundations of data imbalance and solutions for a data democracy. *Data Democracy, Academic Press*, 83-106. https://www.sciencedirect.com/science/article/pii/B9780128183663000058 (01.05.2022)

Lakshmi T C, G. & Shang, M. (2021). *Hands-on Supervised Learning with Python. Learn How to Solve Machine Learning Problems with Supervised Learning Algorithms Using Python*. First Edition. India: BPB Publications.

Landis, J. R. & Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, *33*(1), 159–174.

Leung, K. (2022). *Micro, Macro & Weighted Averages of F1 Score, Clearly Explained*. Micro, Macro & Weighted Averages of F1 Score, Clearly Explained | by Kenneth Leung | Towards Data Science (01.05.2022)

Marvin, L. (2021). *MACHINE LEARNING: Neural Networks, Decision Trees and Support Vector Machine with IBM SPSS Modeler*. Publisher: Lulu.com.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C. & Joulin, A. (2017). Advances in Pre-Training Distributed Word Representations. Facebook AI Research. 1712.09405.pdf (arxiv.org) (01.05.2022)

Millstein, F. (2019). *Natural Language Processing with Python: Natural Language Processing Using NLTK*. https://www.scribd.com/book/431563035/Natural-Language-Processing-with-Python-Natural-Language-Processing-Using-NLTK?fbclid=IwAR3QWtNDqtp_boUh9fB7pnJyJvCYUWfQcmVA6Ow9NogqB_w4blshArQ1yAo (15.05.2022)

Mueller, J. P. & Massaron, L. (2016). *Machine Learning For Dummies*. USA: John Wiley & Sons, Inc.

Patel, S. (2021). *Getting started with Deep Learning for Natural Language Processing: Learn how to build NLP applications with Deep Learning*. First Edition. India: BPB Publications.

Roberts, J. (2017). *Cognitive Behavioral Therapy: How to Rewire the Thought Process and Flush out Negative Thoughts, Depression, and Anxiety, Without Resorting to Harmful Meds.* Collective Wellness Revolution, #1. Isaac Cruz.

Schuyler D. (2013). Arbitrary inference. *The primary care companion for CNS disorders, 15(*3). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3795587/?report=classic (01.05.2022)

Shreevastava, S. & Foltz, P. W. (2021). Detecting Cognitive Distortions from Patient-Therapist Interactions. *Proceedings of the Seventh Workshop on Computational Linguistics and Clinical Psychology*, 151–158. Detecting Cognitive Distortions from Patient-Therapist Interactions (aclanthology.org) (01.05.2022)

Simms, T., Ramstedt, C., Rich, M., Richards, M., Martinez, T. & Giraud-Carrier, C. (2017). Detecting Cognitive Distortions Through Machine Learning Text Analytics. *2017 IEEE International Conference on Healthcare Informatics (ICHI)*, 508-512.

Sochynskyi, S. (2021). *Automated Cognitive Distortion Detection and Classification of Reddit Posts Using Machine Learning*. Tartu University's Institute of Computer Science Master Thesis. Arvutiteaduse instituut - Lõputööderegister

Sullivan, W. (2019). *Deep Learning With Python Illustrated Guide For Beginners & Intermediates: The Future Is Here!*, #2. Healthy Pragmatic Solutions Inc.

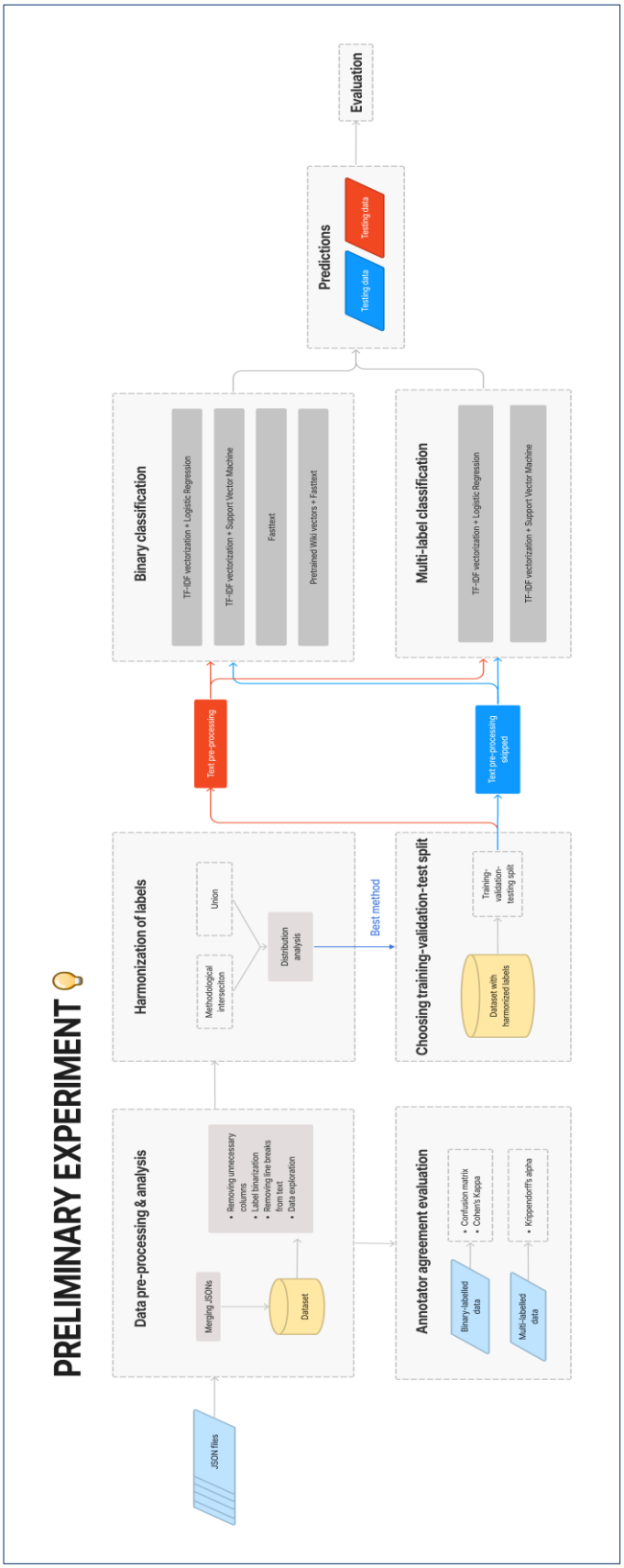
Tausczik, Y. R., & Pennebaker, J. W. (2010). The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Journal of Language and Social Psychology*, *29*(1), 24–54.

Wang, B., Wang, A., Chen, F., Wang, Y. & Jay Kuo, C.-C. (2019). Evaluating word embedding models: methods and experimental results. *APSIPA Transactions on Signal and Information Processing: Vol. 8: No. 1, e19*.

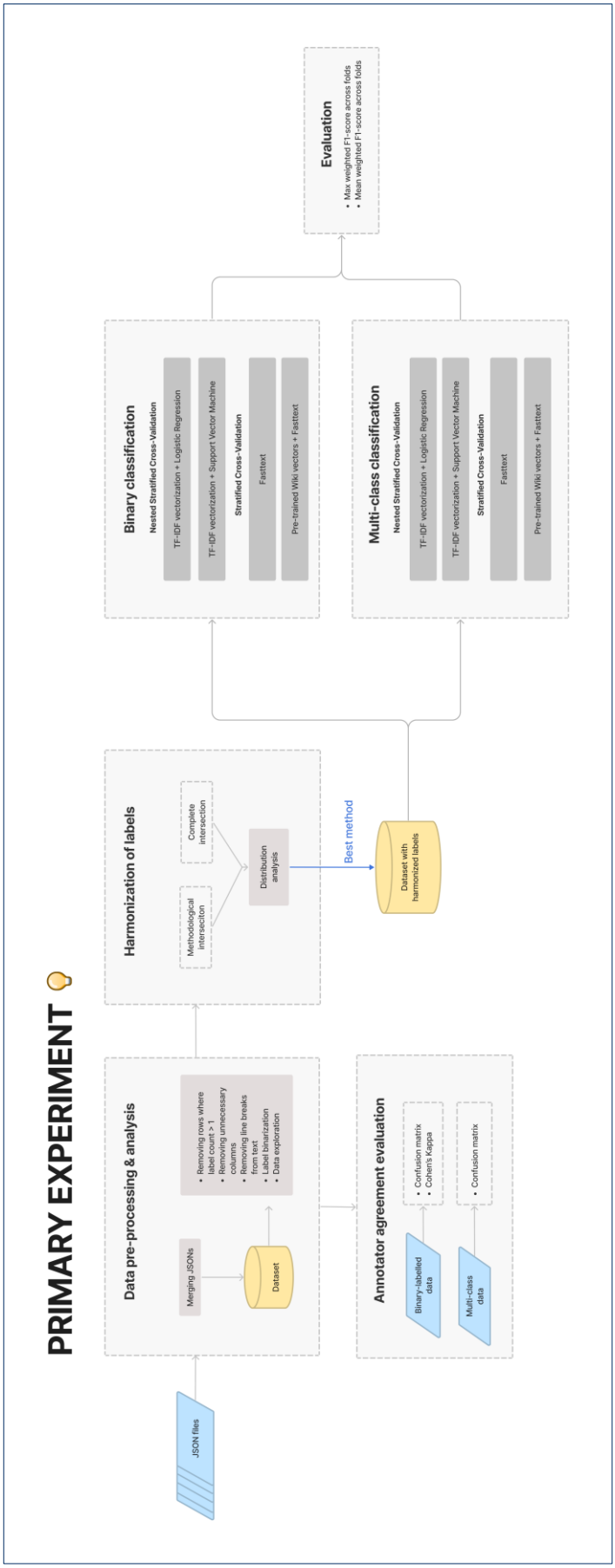
Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory*. Second Edition. New York: Springer-Verlag.

Appendix

I. Preliminary Experiment Method Schema



II. Primary Experiment Method Schema



III. License

Non-exclusive licence to reproduce the thesis and make the thesis public

I, Linda Katariina Grents,
(*author's name*)

1. grant the University of Tartu a free permit (non-exclusive licence) to:

reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis

Predicting Cognitive Distortions from Reddit Posts by Using Supervised Machine Learning Methods,
(*title of thesis*)

supervised by Kairit Sirts (PhD),
(*supervisor's name*)

2. I grant the University of Tartu the permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work from ***17/05/2022*** until the expiry of the term of copyright,

3. I am aware that the author retains the rights specified in points 1 and 2.

4. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Linda Katariina Grents
17/05/2022