

Classification of a
Call For Empathy
in
Child Help Forum Messages

A master's thesis by
Luc Schoot Uiterkamp

May 2021

Committee:

Primary supervisor

Dr. Ing. Gwenn Englebienne

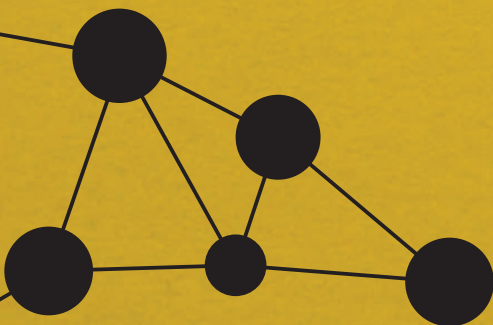
Secondary supervisor

Dr. Hanane Ezzikouri

Additional evaluator

Dr. Mannes Poel

Human Media interaction
Faculty of Electrical Engineering,
Mathematics and Computer
Science
University of Twente



Classification of a Call For Empathy in Child Help Forum Messages.

Luc Schoot Uiterkamp
University of Twente
l.schootuiterkamp@utwente.nl

Abstract

To improve automated detection of empathetic expressions and to streamline online discussion board moderation, an LSTM and a BERT neural network were trained to detect empathetic responses and calls for an empathetic response. Messages from the Kindertelefoon forum, labeled using crowd sourcing, were used as case study to provide a proof of concept. Assessing annotator reliability and determining reply relations were core considerations in cleaning the data. The BERT and LSTM models were trained on empathy detection and on call for empathy detection directly. The empathy detection models were also used in combination with a reply relation algorithm to predict call for empathy. Synthetic oversampling was used to counteract the class imbalance present in the data, as most messages did not contain an expression of empathy. The BERT model performed well in the empathy detection task (MCC = 0.93), the LSTM model did not (MCC = 0.55). The reply relation algorithm was not accurate and neither model performed well on the mediated call for empathy task. The BERT model again outperformed the LSTM model in direct call for empathy classification (MCC BERT = 0.90, MCC LSTM = 0.55). The BERT models perform on par or better than neural networks implemented in empathy classification literature, the LSTM models perform significantly worse. The empathy classification and direct call for empathy classification models using BERT constitute a new state of the art in text-based empathy modelling and text-based emotion classification systems in general.

Keywords

Empathy modelling, call for empathy, Natural language processing, BERT, LSTM

Table of contents

1	Introduction	3
2	Related work	4
2.1	Empathy definitions	4
2.2	Empathy in online contexts	5
2.3	Empathy for youths versus adults	7
2.4	Forum post hierarchy	7
3	Language model background	8
3.1	Word representation	8
3.2	Text representation	9
3.3	Embedding	9
3.4	Imbalanced data	9
3.5	Models	9
3.6	Model comparison methods	15

4	Dataset background	15
4.1	Kindertelefoon forum website functionality . . .	16
4.2	Data descriptives	16
5	Methods	17
5.1	Interview	18
5.2	Relations between posts	19
5.3	Features	21
5.4	Models	22
5.5	Data collection	23
5.6	Annotation	24
5.7	Data processing	26
6	Results	27
6.1	Annotators and agreement	27
6.2	Dependency builder	28
6.3	BERT pretraining	29
6.4	Empathy detection	30
6.5	Call for empathy detection	36
7	Discussion	43
7.1	Annotation agreement	43
7.2	Reply relations	43
7.3	BERT pretraining	44
7.4	Empathy classification: BERT	44
7.5	Empathy classification: LSTM	45
7.6	Call for empathy	46
7.7	Oversampling	46
7.8	Relation to related work	47
7.9	Limitations	47
8	Conclusions	48
9	Summary	48
10	Implications and future work	49
11	Acknowledgements	49
A	Interview setup	52
A.1	General questions	52
A.2	Focus group Kindertelefoon	52
A.3	Interview psychologen	53
B	Empathy components overview	54
C	Agreement scores full figures	55
D	Annotation applications	57
E	Activation functions	59
F	Top 30 antecedent label and reply resolution prediction counts	60

1 Introduction

All layers of our society spend an increasing amount of time online, as more people work from home and online education rises quickly¹. Our social lives also increasingly take place online, on internet fora and social media. Managing the stream of data which is produced by our increasingly online lives is primarily the responsibility of the platform used (Renda, 2018), but managing it by hand becomes unfeasible as the datastream increases in size.

To automatically moderate user-generated messages, an algorithm which can extract meaning from them and interpret this meaning can be used. Deriving literal and emotional meaning from written messages is not difficult for most humans, but the creative manner in which language is written makes it difficult for computers. This is especially true for emotional meaning, as there are no set-in-stone rules which can be applied to determine whether a text expresses for example happiness, sadness, empathy or apathy. With machine learning algorithms, patterns in these texts which are much more complicated than can be expressed in rules can be found and used to infer some amount of sentiment from texts.

Apart from managing misinformation and disinformation², platforms which wish to create a safe and supporting environment for their users have to manage very subtle variations in content. This nuanced moderation makes it much more difficult for algorithms to distinguish between acceptable and unacceptable content, as sentiments used for these models often represent extremes on a scale, such as positive or negative opinions regarding a certain subject or a happy/angry distinction. Communities which target vulnerable audiences such as children face these difficulties, as they not only have to prevent and identify abuse and bullying but also provide a space in which children feel safe to express their story, questions and worries.

Identifying which messages and or users may be prone to abuse and which messages express abuse is such a more nuanced content interpretation, and this project concerns the detection of messages which express such a vulnerability to abuse. These vulnerabilities are operationalized as messages to which an empathetic response is appropriate, as such a ‘call for empathy’ requires a the user to expose certain vulnerabilities. To establish a context on interpreting natural language texts and to define what the concept of empathy means in the context on online fora, several related works are discussed in section 2.

Text messages from a forum aimed at supporting children, managed and hosted by the Kindertelefoon, were used as a case study to train and test the models. The Kindertelefoon is a Dutch volunteer organisation aimed at giving children between ages 8 and 18 a place to talk about problems at home or school, about mental or physical health or any other topic they like. This data was found to be suitable as this forum faces the moderation difficulties in creating a safe space for their users as its target audience are children and parental supervision when visiting the forum is often absent due to the nature of the forum.

To be able to interpret content to moderate messages in order to provide such a safe space, an accurate language model is necessary, not only in the default language in that space, but of the audience and topic-specific language used. Children have a different vocabulary from adults, and use different words, sentence

structures and narrative structures to convey a message. These differences need to be incorporated in order to be able to interpret a nuanced distinction between a toxic message and an acceptable one. The platform-specific structure should be incorporated in this model as well, which enables platform-specific features to be used in modelling, such as topics, tags, titles, awards or roles.

Within threads, the relation between responses needs to be known in order to be able to interpret for example toxicity or abuse: a given message might be acceptable in one context but not in another. These response relations are also necessary to infer the post calling for empathy from an empathetic message. As they are not represented on the Kindertelefoon forum, these relations need to be established, and an algorithm to infer response relations was developed. The process of developing this algorithm is described in section 5.2.

The messages are downloaded from the public forum, cleaned, and stored in a systematic manner. To be able to make sense of the data, it needs to be annotated, for which a website was built. The process of collecting data and developing the means to annotate it are described in section 5.5. As many different annotators reviewed different parts of the data and the annotators were of varying reliability, a program to assess annotator quality was developed. This program was used to determine which annotators performed well enough for their annotations to be used in the final dataset.

To automate the detection of call for empathy posts, two models which aim to classify messages which call for an empathetic response are developed and tested. As the concept of a call for empathy is more difficult to define than the concept of empathy, these call for empathy posts are identified through their empathetic responses. Messages which express empathy are classified first, and through determining to which posts these messages are a response, posts which call for empathy are classified.

As an accurate language model of the specific language used in the data is already made for the empathy classification algorithm, a classifier which directly classifies the call for empathy without reply relation inference is also developed. This direct model serves to explore the abilities of the language models made in this study in understanding the implicit cues given in the call for empathy posts. If this direct model is able to achieve this, it will do so faster, as it is not reliant on replies to a call for empathy post for classification, and possibly more accurately as there is no intermediary step of determining reply relations. The functioning and implementation of both the empathy classification models as well as the exploratory model is described in section 5.4.

The performance of the developed models is described in section 6, where the annotations are used as a reference for the model performance. The performance is discussed in section 7, as are the comparisons between the models and the reflections on the study in general.

¹Because of the Covid 19 pandemic, currently all education is online, but online tertiary education is becoming increasingly common.

²The distinction being the intentionality of the spreading of false information, see Renda (2018).

The following enumeration details the five core research questions (1-5) with their respective subquestions (a, b) along with a summarized answer approach marked with →.

- (1) Which annotators are reliable and which are not?
→ Annotator reliability assessment (section 5.7).
- (2) How well can (the different components from) the reply relations algorithm assess the reply relations?
→ Assess accuracy of the (sub)model(s) (section 6.2).
- (3) How well do the LSTM and BERT model classify empathy?
→ Use MCC score and model loss to evaluate what the performance is and if there is a scope for improvement (section 6).
 - (a) How does BERT pretraining impact this?
→ Compare different pretraining epochs (section 6.4).
 - (b) How do trainable transformer layers in BERT impact this?
→ Compare different pretraining epochs for a model without trainable transformer layers (section 6.4.1).
- (4) How well does the combination of empathy prediction and reply relation work?
→ Assess MCC and accuracy of the combined empathy prediction and reply relation (section 6.5.1).
- (5) How well do the LSTM and BERT model classify call for empathy directly?
→ Use MCC score and model loss to evaluate what the performance is and if there is a scope for improvement.
 - (a) How does BERT pretraining impact this?
→ Compare different pretraining epochs (section 6.5.2).
 - (b) How do trainable transformer layers in BERT impact this?
→ Compare different pretraining epochs for a model without trainable transformer layers (section 6.5.3).

A number of significant contributions to the field of sentiment analysis and text mining are presented in this work. The annotator quality algorithm, which assesses annotator quality in a setup with many annotators, enables quality control in crowd-sourced annotations without manually inspecting very large datasets. The reply relations algorithm, though in need of parameter optimization for improved accuracy, provides a foundation of six components which can be used to determine post relations in forums which do not encode this natively, increasing the richness of a scraped dataset from such websites. The empathy classification language models (specifically the BERT model) and the BERT call for empathy classification model represent a large step forward in classification of empathy through computer models.

2 Related work

2.1 Empathy definitions

The concept of empathy, although intuitively familiar, is ambiguous and has been described in various ways. A commonality between these descriptions is the description of an insight in another person's emotions but this insight is expressed differently in the several definitions. In the *Social Psychology* textbook, empathy is described as "a cognitive component of understanding the emotional experience of another individual and an emotional experience that is consistent with what the other is feeling" (Kassin et al., 2019, p. 398). These two components are often at the core of

the definition for empathy (Batson, 2009; Cuff et al., 2016; Decety & Jackson, 2004; Spencer et al., 2020)

The cognitive component defines the amount of insight that is had on the context and circumstances and the impact that events have had on another person. The emotional component defines the ability to imagine what those impacts and circumstances feel like to that other person. Other sources include, apart from these two components, an appropriate compassionate response to another person's feelings (Levenson & Ruef, 1992). This can be seen as the operationalization of empathy. These operationalizations are often rooted in imitation (Iacoboni, 2005; Kassin et al., 2019; Pfeil & Zaphiris, 2007). This imitation helps understand an experience of an other person by literally copying it and conveys to the other person that a similar feeling is felt. This can be expressed in similar language, stance, facial expressions and intonation.

A definition derived from a study observing both empathizers and targets for empathetic responses (Håkansson & Montgomery, 2003) defined four major constituents of empathy, which need to all be present in order for an interaction to be marked as 'empathetic'.

- (1) The empathizer understands the target's situation and emotions
- (2) The target experiences one or more emotions
- (3) The empathizer perceives a similarity between what the target is experiencing and something the empathizer has experienced earlier
- (4) The empathizer is concerned for the target's well-being.

Batson has collected and summarized eight definitions of empathy as used in psychological literature (Batson, 2009), which give a narrower distinction between several different ways of defining empathy. In these eight concepts, the cognitive, emotional and compassion components are found in varying degrees, as well as the four constituents from Håkansson and Montgomery. Although the concept of empathy is generally considered to be more than one of these components (Decety & Jackson, 2004; Spencer et al., 2020), it is useful to distinctly define facets of empathy so that nuances in different age groups may be identified as is important in this study and so that a well-grounded and complete definition of empathy may be constructed. The eight definitions by Batson are used as guidelines in defining the several aspects of an empathetic response.

2.1.1 Knowing another person's internal state

The first definition of empathy is a cognitive one and as such is also known as 'cognitive empathy' or 'empathetic accuracy'. Defining empathy as knowing another person's internal state refers to being aware, through linguistic or nonverbal communication, of what is on the other person's mind. This notion is the first of Håkansson and Montgomery's constituents of empathy. It may not be accurate or complete, this definition merely requires an active awareness of one person's belief of another person's internal state.

2.1.2 Physical mimicry

A more neurological perspective of empathy is based in physical mimicry. This view denotes that empathy is gained from purposeful simulation of another person's (facial) expression or that empathy necessarily coincides with neurological and physical mimicry (Niedenthal et al., 2010). The core concept in this perspective is that the embodiment of an emotion causes neurological pathways to activate similarly to the way they would

if the person was the primary experiencer of the emotion. This gives an impression of what another person is feeling through the vicarious experience.

2.1.3 Feeling how another person feels

An affective perspective of empathy is coming to feel as another person is feeling. This is more than merely knowing another person's internal state and requires more than merely physical mimicry. This definition is based on *experiencing* the emotion that another person is having. This concept of feeling how another person feels is usually known as empathetic contagion (Calvo et al., 2015) or outside psychology as sympathy (Batson, 2009).

2.1.4 Imagining how another is thinking and feeling

Although seemingly similar to the first (cognitive) definition, imagining how another is thinking and feeling extends merely concluding how the other feels with imagination based on what is known from previous experiences with that person or with other people. This is not necessarily based on one's own experiences or character but rather on what the perspective taker thinks the other person experiences.

2.1.5 Literally perspectivising

A somewhat archaic but still well-known perspective is literal perspective taking. Here, one tries to not only take perspective in the situation of another person but also to reason the way that person would reason. This involves an extensive perspective taking ability to the point in which it is unreasonable to assume this approach might be actually feasible. Rather, the core principal is to get as many contextual factors correct in empathising.

2.1.6 Imagining how one would feel in another person's place

A view which is often referred to by the term 'perspective taking' is to imagine how one would behave and feel in another person's place. This is different from imagining how another is thinking or feeling and from literally perspectivising as imagining in place is based on one's own experiences and character in another person's situation instead of the other person's character. This is the third constituent of Håkansson and Montgomery's study (Håkansson & Montgomery, 2003). The active reflection on one's past experiences contributes to the connectedness with another person, as commonalities are sought which may shed light on how one would act or feel in another's place (Spencer et al., 2020).

2.1.7 Feeling distress because of another person's malaise

Distinct from feeling distress *with* another person because of perspective taking, empathy as feeling distress as a result of witnessing another person's suffering has also been used as a definition of empathy. This concept is also known as 'empathetic distress'.

2.1.8 Feeling for another person's suffering

A perspective based in a more altruistic sense than the other definitions, empathy is also defined as feeling distress or discomfort because of another person's distress. This perspective is different from *feeling how another person feels*, as the reactionary emotion does not need to be the same. This is the forth constituent of empathy according to Håkansson and Montgomery.

2.2 Empathy in online contexts

In a face-to-face, offline context, people use non-verbal signals as well as literal verbal expressions to express empathy. For example, a hand placed on a shoulder, facial expressions and intonation are used to convey empathy alongside literal expressions (Eisenberg et al., 1997). In general, non-verbal channels make up around 90% of emotional expressions (Goleman, 1995; J. J. Preece & Ghazati, 2001, cited in Pfeil and Zaphiris (2007)). Given the lack of these non-verbal communication channels in the forum, users are completely reliant on literal expressions and replacements for non-verbal expressions in the form of emoji's and similar expressions of feeling. Several studies have found that the lack of non-verbal language channels has a much smaller influence on the presence and experienced reception of empathy in online communities than the type of community and the gender ratio have. Support fora and online communities with a relatively large amount of women have a larger amount of empathetic responses than other types of fora such as cultural or religious fora or fora with a larger ratio of men (Garcia-Pérez et al., 2016; J. Preece, 1999; J. J. Preece & Ghazati, 2001).

In their study on virtual empathy in the context on online teaching, Garcia-Pérez et al. find that empathetic stress (concept 7, section 2.1.7) and the adoption of perspectives (concept 6, section 2.1.6) are particularly important for online communities in which users feel safe, motivated and in which positive relations can be had (Garcia-Pérez et al., 2016).

According to Caplan and Turner (2007), three conditions must be met in an online environment for that environment to be comforting to its users and conducive of empathetic responses from peers.

- (1) Participants must be willing to enter into a conversation that will involve discussing upsetting matters
- (2) Conversation must be focused on the distressed individual's thoughts and feelings about the upsetting experience
- (3) The distressing matter must be discussed in a way that facilitates reappraisals

The third item in this list may be achieved through expressing thoughts into a narrative, thereby structuring it and putting ideas in words. This encourages reflection, through which the act of writing down thoughts and feelings into a story may promote positive reappraisals and lead to an improved affect state (Caplan & Turner, 2007). In the Kindertelefoon forum, this narrative structure of posts defines the 'emotional vent' type of post. This confirms that this type of post is indicative of a 'call for empathy' post. An environment in which these posts can be placed without fear of exposure or harassment is created on the Kindertelefoon forum because it is heavily moderated (Garcia-Pérez et al., 2016), anonymous, and because 'troll' comments or off topic comments are frequently altered or deleted by the Kindertelefoon moderators.

Pfeil and Zaphiris (2007) have studied patterns of empathy in online interactions on a discussion board on the SeniorNet platform, where elderly can find information, news and contact with other elderly. The study used a discussion board on depression within the SeniorNet platform, analysing 400 messages from the board. The messages were coded into 23 codes in 7 categories. The empathy-related codes of the 23 codes in total consist of

target-related³ and empathizer-related⁴ constituents of empathy, which are indicative of which members play what role on the forum. For example, the *Ask for support* code is indicative of a call for empathy, whereas the *Similar situation* code indicates an empathizing role although this in itself can be responded to with an empathetic response. This unclear distribution of empathizer and target roles is found in the SeniorNet study, and is distinct from what would be expected in an offline scenario.

The following constituents were identified as important parts of online empathy on the SeniorNet forum.

Understanding, although this is not explicitly described often, which might be due to offline understanding often being non-verbal or contextual. A quizzical look or an affirming nod often fulfills this role and this it is difficult to find an online alternative. Understanding is an important differentiating factor between light support such as phrases like ‘hang in there’ and deep support, which is more personalized and specific to the situation that the target is in.

Emotions, both from the target as well as the empathizer’s perspective featured more prominently than factual information. Especially negative emotions functioned as a call for empathy from the side of the target, which were often met with both positive and negative emotions from the empathizer side.

Similarity, in the sense that empathizers have experienced or are aware that they can easily experience a similar situation is an important aspect of empathy and sympathy in offline conditions. This was expressed in the SeniorNet study as well, indicating to empathy targets that they are not alone and that others know what they are going through, that others share their story and that others are there for them to help *because* they know what they are going through.

Concern and caring for others are very personal properties, which are expressed in the SeniorNet data through specific references to others in regards to how they are doing. This is differentiated from other expressions of empathy by the fact it was initiated by the empathizer and not by the target. This indicates personal concern and involvement.

Coulson (2005) studied an online support group for people with irritable bowel syndrome. They hand-analysed messages and labeled them with the labels ‘emotional support’, ‘esteem support’, ‘information support’, ‘network support’, and ‘tangible assistance’. They found that users often vented their frustrations online if things were not going well. These kinds of posts were usually met with empathetic responses providing emotional support. Emotional venting posts as they are present in the Kindertelefoon data may also be expected to call for empathy. In esteem support responses, users compliment others in their ability to cope with difficulties. Similarities between the situations in which the target and responder are situated are emphasized if present, which expresses empathy.

Spring et al. (2019) distinguish three emotion detection strategies: rule based, non-neural network and deep learning. Rule based approaches simply match responses to keywords which indicate certain emotions. This may be extended to for example n-grams. This simple mapping is not robust and is highly dependant on the keys with which the responses are compared. For the Dutch language, no empathy-specific lexicon exists. Therefore, rule based methods are not considered in this study. The

second strategy of emotion classification is through a non-neural network classifiers such as support vector machines, decision trees or naive bayes classifiers (Spring et al., 2019). An example of such a study is one in which a support vector machine classifier is used to detect empathy in counseling by Xiao et al. They used human-labeled transcriptions to train an n-gram based support vector machine classifier (Xiao et al., 2015). One major advantage mentioned of a classifier which is able to classify the use of empathy in natural language is that it is able to give immediate feedback to the counseling process. The counsellor can use feedback to adjust their attitude. Similarly, the algorithms developed in this project can be used to not only alert a human to an empathy-requiring forum post but can also provide feedback on the appropriateness of a response.

In their study, automatically generated transcripts were used and were found to be fairly accurate. These transcripts were annotated by human annotators, which yielded a dataset of counselling session text labeled with high/low empathy labels. From these documents, n-grams (n=1, 2, 3) were derived which were smoothed with Kneser-ney smoothing. These n-grams were used to train a support vector machine classifier, with which new texts can be automatically labeled. Xiao et al. observe that n-grams indicative of the high-empathy class are often expressions which indicate reflection, while n-grams of the low-empathy class relate to probing for more information or giving concrete instructions (Xiao et al., 2015).

An n-gram solution to detect phrases which are common in phrases expressing empathy is problematic in the context of this project, as the number of misspellings and varying grammatical structures yields unreliable n-grams and sparse representations. A severe filtering and error correction can mitigate this problem partially. Such filtering can consist of a simple stemming or a full spell-checker, which may improve the accuracy at the cost of interpretation accuracy. Despite this improvement, the varying grammatical structure is still problematic for n-grams with n more than 1.

The third strategy defined by Spring et al. is to classify emotion through neural network models. They mention LSTM models as a good candidate models for such tasks, which is used in several studies (Feng et al., 2019; Khanpour et al., 2017).

The LSTM model used by Khanpour et al. (2017) uses convolutional and LSTM layers to process messages from a lung cancer discussion board on the Cancer Survivor’s Network. The ConvLSTM model was compared to rule-based approaches (the first of the strategies defined by Spring et al.) and was found to outperform them significantly. The convolutional layers in the model were implemented to achieve dynamic embeddings of the input, after which LSTM layers were used, in combination with a softmax activated fully connected layer to produce the output classification.

In general LSTM models are popular in sentiment and emotion classification tasks. However, since Google researchers published the BERT model, this architecture has been used in emotion classification and sentiment analysis as well. Although BERT has not been used for empathy classification yet, it has proven to perform well on similar sentiment and emotion classification tasks.

Sun et al. (2019) compare LSTM and BERT models in sentiment classification. They use aspect based sentiment analysis, splitting different sentiment utterances in a text along the aspects they evaluate (Saeidi et al., 2016). For example, in considering reviews

³General feeling, Narration, Medical situation, and Ask for support

⁴Interest, Encouragement, Best wishes, Deep emotional support, Reassurance, Give help and Similar situation

of products with aspects which are positive and which are negative, these aspect based models should be able to disambiguate which aspect is positive and which is negative. A biLSTM model outperformed the BERT model when it was trained on raw data but with preprocessing, BERT outperformed the biLSTM model. This preprocessing consists of feature extraction from the input sentences. For the Kindertelefoon data, feature information apart from the raw text is available which could be applied in a similar way. As aspect based analysis is too complex for this study, the assumption is made that messages either contain an empathetic expression or do not and that each message has only one target.

Li et al. (2019) also compare LSTM models to BERT models in the context of sentiment analysis. Like Sun et al. (2019) they use BERT as an encoder and compare several classification output layers which produce a class prediction. The datasets used are reviews of consumer products, specifically laptops, and of restaurants. The simplest linear output layer already outperformed LSTM models, but the BERT model using a gated recurrent unit layer and the model using a self attention layer performed best on the laptop data and restaurant dataset respectively.

Other constructions using BERT as encoder are also used in emotion classification. In Yang et al. (2019), multiple utterances were concatenated into one input, separated by [SEP] tokens. This was possible because of the small size of the documents. The output of the BERT layers are separated along the input [SEP] tokens. The output sections were pooled using max pooling, and subsequently classified per utterance. This setup benefits from very fast training because of the simultaneous processing of multiple documents. However, due to the document length in the Kindertelefoon data, this is not possible in this study.

2.3 Empathy for youths versus adults

On the Kindertelefoon forum, the expressions of empathy are posted by both teens and Kindertelefoon volunteers. Several studies indicate that empathetic skills are still in development in the age range in which the teens which visit the forum are, although variations exist in the extent with which and exact age range in which these developments occur. In a review of studies on empathy development in adolescents (age range 11 - 18), Silke et al. (2018) have found a variety of operationalizations of empathy. For example, a number of studies chose to only investigate the affective aspects of empathy while others limited themselves to the cognitive aspects.

Stern and Cassidy (2018) summarize earlier work (Eisenberg, 2000; Hart & Fegley, 1995) in their claim that sociocognitive developments during teen years correlate with the ability to empathize through the improvement of theory of mind of others, emotional understanding of others and emotional self-regulation and self-awareness. Haugen et al. (2008) similarly cite others in their hypothesis that empathetic accuracy should increase as teenagers grow up, as increasing cognitive and emotional skills facilitate a better insight into the emotional state of others, which include better perspectivising, verbalization and abstract thinking. However, they were unable to find a correlation between empathetic accuracy in adolescents between 14 and 19.

Eisenberg et al. (1997) observe that the detection of non-verbal expressions of empathy is still in development in teens, which makes them more reliant on more verbose expressions of empathy, which is consistent with the perspective drawn by Haugen et al.

In their review of studies on perspective taking and altruism, Underwood and Moore (1982) indicate that the increased ability of role-taking is a development which is necessary for the development of empathetic perspective taking.

Several studies have found the development of empathy to be moderated by the gender of the teenagers under study. Van Tilburg et al. (2002) find that there is a strong effect of age on empathy between ages 11 and 14 but only for girls with only a weak effect for boys. Kalliopuska (1983) found in an evaluation of empathy among school-aged children that while girls in general have a higher empathy score as measured with a self-report and a peer-report questionnaire, empathy scores did increase with age between ages 11 and 19. In a neurological study on gender differences, Christov-Moore et al. (2014) cites many sources which indicate a higher empathy among adolescent girls is higher than among boys the same age. In a review of factors influencing empathy development among adolescents, Silke et al. (2018) cite many studies which have found the same result.

The development of empathetic skills in the forum users' age bracket can be expected to be varied. The forum offers by its nature a more verbose expression of empathy than real life, which alleviates part of the possible lack of empathy expression or sensing skill users might have. The works reporting on empathy development are not conclusive nor concrete enough to warrant a differentiation on the empathy concept based on this aspect.

2.4 Forum post hierarchy

As the call for empathy posts are primarily classified through the detection of empathy-providing posts, the relations between the posts need to be mapped. In general, the disambiguation of relationships between posts on internet fora is useful in a number of ways. For example, it enables large datasets to be mined for natural language research and discourse analysis. The mapping of discourse structure is necessary as is often not encoded on the online resources themselves (El-Assady et al., 2018). An accurate mapping of relationships between posts also helps online resources themselves, for example to understand what answer was given to which question in help-seeking fora. This helps future users find an answer to their similar question quickly. It may also be used to determine when a thread should be considered 'stale', a state in which no new useful answers are likely to be posted. In this latter goal, disambiguating relationships of posts is often paired with dialogue act labeling, in which posts are labeled by their role in the forum thread (Kim et al., 2010). For example, if in a given thread on a technology help forum there are many posts which indicate a similar problem but no posts occur which provide a solution, the thread may be marked as stale and closed or alternatively may be marked as important for more users with potential answers to see. In order to determine which post relates to which other post, several heuristics and algorithms can be employed.

Xi et al. developed a method to yield concise search results for a given query from discussion boards. In developing this, they have defined five relationship types which a post on a discussion board can assume. These relationship types help group conversation threads from within a larger thread and are listed in figure 1. From these relationships, the question and answer relationships are complementary. The agreement/amendment relationship, the disagreement/argument relationship and the

1. Question relationship: a user may not be clear about the information in the previous message(s) and as a result, raises more questions in the replied message. This type of relationship is a very good indication of a shift in topic.
2. Answer relationship: Current message answers the question of the previous message(s). [...]
3. Agreement/Amendment relationship: In the replied message, user expresses their agreement or adds amendment to the information presented in previous message(s). [...]
4. Disagreement/Argument relationship: In the replied message, user expresses their disagreement or argument to information presented in previous message(s). [...]
5. Courtesy relationship: "Thank you", "You're welcome" messages. [...]

Figure 1: Five types of relationships posts may have according to Xi et al. (2004)

courtesy relationship may refer to a statement, answer or question and are therefore not as clear-cut as the question-answer pair.

In their papers, Shrestha and McKeown (2004) and Cong et al. (2008) describe several approaches for classifying sentences as questions. The easiest approach is to use regular expressions to identify question marks and keywords which indicate questions, such as what, who, where, why, when and how. This approach is easy to implement but fails to detect questions in a declarative form such as 'I would like to know how deal with this.'. Additionally, question marks may be used to express uncertainty instead of a question, which leads to false positives.

Shrestha and McKeown (2004) propose using part of speech (POS) tags for the text to classify. Each sentence is tagged and the first and last five POS tags are used to classify the text. In their comparison with manually annotated data, they found that this method works well for interrogative questions but still not for declarative questions.

Cong et al. (2008) combine keyword detection with the method proposed by Shrestha et al. and encode texts such that all but the keywords are POS tagged. This yields a text encoding which looks as follows: 'where, can, <PRP>, <VB>, <DT>, <NN>'. They then used n-grams ($n=1-2$) of this data to train a classifier. They compared their approach with those of Shrestha and McKeown and a keyword detection approach and found significantly better results in their approach with F_1 scores of 0.24, 0.86, 0.84 and 0.97 for keyword detection, question mark detection, the approach by Shrestha and McKeown and their own approach respectively.

To pair the appropriate answer to the detected question, Shrestha and McKeown use a similarity score, under the assumption that an answer uses the same vocabulary as the question. Cong et al. take this basis but improve it with features from the forum such as reply distance.

The relationships from figure 1 which are less clear-cut than question-answer pairs may be uncovered by similarity, heuristics and meta features (features which are not post content). In recovering thread structure from discussion fora in which thread structure is not represented in the website, Y. Wang et al. (2008) use cosine similarity to detect posts which use similar language. From this, they compose a graph of post responses with similar language use and restructure the thread in correct response order.

Many models, such as the ThreadReconstructor by El-Assady et al. (2018) but also the studies by Kim et al. (2010) and Aumayr et al. (2011), use features which are not part of the posts' content and may be specific to the dataset in question to help determine post relations. This may range from time distance, post distance or different authors (Aumayr et al., 2011; El-Assady et al., 2018), to the number of question marks, exclamation marks and URLs in a post, or even user profiles with information of which type of post is often posted by that user (Kim et al., 2010).

3 Language model background

The following sections provide background information on several aspects of language modelling relevant for this project as described in previous works.

3.1 Word representation

In an n-gram word representation such as the one used in Xiao et al. (2015), each text is encoded as a series of word patterns. These word patterns are called n-grams and may have differing lengths n . For example, a common $n = 3$ n-gram (also called a *trigram*) in this text is "call for empathy". N-grams are derived from a corpus of texts in which every word combination of n words is counted. The final n-gram set usually only includes n-grams with a minimal frequency number in order to reduce the number of n-grams which are used to encode texts. These text encodings can be used in a classification algorithm by calculating the Maximum Likelihood Estimate (MLE) for each class for new combinations of n-grams in new documents, but they can also be used as features in other models. In the coming section, n-grams are considered as features for a model and not as a standalone MLE classification method.

N-grams enable localized context to be used, as they encode a section of text instead of a single word. These contexts are limited however, as n-grams for $n > 3$ rarely improve performance over uni-, bi- and trigrams. N-grams with a large n also get increasingly rare in texts because of the lower probability of any n words occurring if n is large. For example, the $n = 5$ n-gram 'The BERT model performed better' has only one occurrence in this text, whereas the $n = 3$ n-gram 'The BERT model' occurs 68 times.

The expected grammar and spelling inconsistencies in the Kindertelefoon data make an n-gram representation a poor choice, as misspellings, uncommon contractions and loanwords yield many unique n-grams. Additionally, grammar mistakes increase this problem for n-grams with $n > 1$ as grammar mistakes yield unusual contexts. Like word-based n-grams, character based n-grams can be constructed. Character n-grams do not suffer from the downsides caused by grammar and spelling inconsistencies as much, as they are concerned with much smaller pieces of text. However, character n-grams lack the ability to use even localized context for the same reason.

A wordpiece representation such as the one presented in Senrich et al. (2015) can represent words in a vocabulary like uni-grams but can additionally subdivide unknown words into character n-grams. This gives the model the best of both worlds: a representation of whole words if the word is known and word pieces if the full word is not in the vocabulary. This enables a model to use information from a part of the word if the full word is unknown but also from a misspelled word, without the need for extensive preprocessing of the data, during which information is lost. A wordpiece representation can for example divide a

word such as the misspelled word “misrepresenting” into “mis” + [UNK] + “##ing” which captures valuable meaning from the word despite the word not being in the vocabulary. The prefix ‘mis’ is indicative of a negation and the ending ‘ing’ indicates that the word is most likely a verb or a noun. The context of the sentence can provide more evidence on which of the two it is.

3.2 Text representation

Regardless of whether the input features for a machine learning model are plain tokenized texts, n-grams or wordpiece word representations, they need to be organised in a specific way. For different models, optimal text representations may differ, as is the case for the two models which are used in this study.

Many models use a ‘bag of words’ representation, which is constructed in terms of the vocabulary of a model. Each vector representing a text has a dimension for every word in the vocabulary, in which the frequency of that word in the text is encoded. Large vocabularies enable very diverse texts to be represented accurately without gaps but yield sparse representations. Small vocabularies consisting only of more common words yield less sparse representations but leave more gaps in the text representation because of words missing from the vocabulary. A major disadvantage of the bag of words representation is the loss of word arrangement, as all texts are encoded as frequencies of the vocabulary items. Additionally, as the representation is based on the frequency of the terms, words with a high prior probability weigh in more than words with a low probability, even if words with a low probability might be more informative.

The TF-IDF representation (Term Frequency Inverse Document Frequency) takes word rarity into account by multiplying the term frequency with the *log* of the inverse document frequency. This yields a combined score of how many times a word is featured in one document in relation to how frequently it occurs in all documents. The TF-IDF formula is shown in equation 1, in which $tf_{t,d}$ is the raw term frequency of term t in document d , N is the total number of documents and df_t the number of documents in which the term occurs.

$$TFIDF_{t,d} = tf_{t,d} \times \log \left(\frac{N}{df_t} \right) \quad (1)$$

Alternatively, texts can be represented in the original order, indicating not the frequency but the vocabulary index in each position. This yields a representation in which the order of original text remains intact, which is valuable information discarded in TF-IDF and BOW methods. Representations in which the words are presented in the original order must have some other way of mapping the vocabulary to the input, and can additionally not access frequency data for important words directly, though this can be inferred.

3.3 Embedding

There is no inherent meaning in TF-IDF or BOW text representations, which is why these representations are used in combination with an embedding layer. Usually, these embedding layers are trained to represent words with vectors in such a way that similar words yield similar word vectors. For example, the embedding vector for the word ‘king’ will be similar to the word ‘queen’ but also to the word ‘man’, though they will be similar in different dimensions of the embedding. These similarities are based on co-occurrence, which is a feature which is naturally represented

in frequency based representations. This embedding is based on the assumption that frequently co-occurring words are similar words. After training, the word embeddings are simply saved in a lookup table with the original words. These word embeddings are the same for each occurrence of the word, regardless of context in the sentence.

The Embeddings From Language Models (ELMo) word embedding is more dynamic. Instead of training static vectors, ELMo embeddings consist of trained functions of hidden states in the model it is applied in. This means that the embedding for the same feature can differ based on surrounding features, although the embedding will be at least somewhat similar regardless of context. In LSTM models, for which ELMo is best suited, this means that the embedding for the input in each step is dependant on the previous step.

Transformer models produce similar context dependant embeddings, as this is the core of the encoding part of the model. The lack of innate sequentiality in transformer models enables them to use truly bidirectional context in these embeddings, instead of only using past information as is the case in ELMo.

3.4 Imbalanced data

As the proportion of texts containing empathetic expressions is small, the dataset will be imbalanced, which has large consequences on training and evaluating the models. There are two principal methods of coping with class imbalance: oversample the minority class (or undersample the majority) or incorporate the class imbalance in the model.

The simplest way to balance classes is to undersample the majority class until the classes are balanced. As this removes a lot of training data from the model, this is undesirable. Duplicating texts from the minority class until the classes are balanced does not cost information, but does not add information to the model either. Additionally, since minority samples may be duplicated many times before the classes are balanced, this oversampling technique is prone to overfitting the minority class data and as a consequence poor performance on real world data.

The Synthetic Minority Over-Sampling Technique (SMOTE) (Chawla et al., 2002) uses the average of K nearest neighbors of texts in one class to synthetically produce new examples. This only works if the features are in a space in which the scale is meaningful and not in a categorical space in arbitrary order. This means that a feature embedding such as Word2Vec should be used to produce averages, and not a vocabulary encoding such as bag of words.

The other principle method for coping with class imbalance is to incorporate it in the model. For machine learning models, the loss function can be altered to be more punishing when minority texts are misclassified, as was implemented for BERT by Madabushi et al. (2020). They customized the BERT loss function by multiplying it with a label dependent weight. Though the results are promising without a need for synthetic data, this technique is not as tried-and-true as synthetic oversampling and is therefore not applied in this study.

3.5 Models

A large range of models which are able to process text in some form have been developed over time. The models highlighted in the following section are therefore not an exhaustive list but are meant as an insight into state of the art models which are relevant to the subject of emotion classification in natural language.

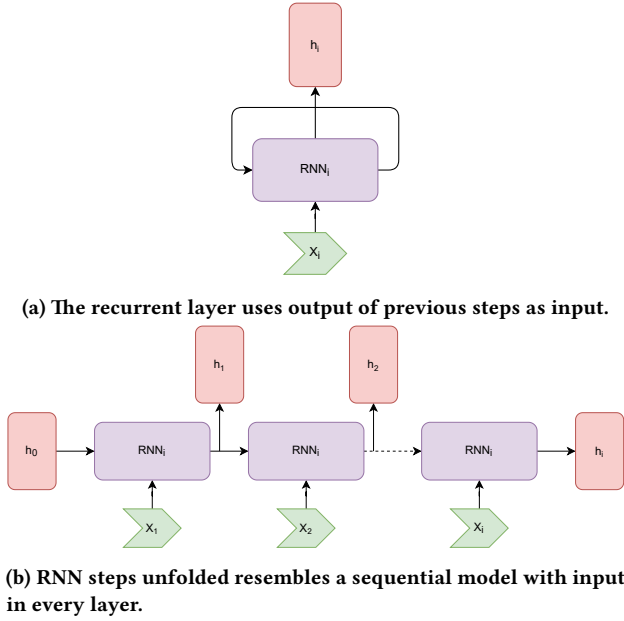


Figure 2: Visualization of a recurrent neural network. X_i denotes input at timestep i , h_i the hidden state and RNN_i the recurrent layer with parameters in timestep i .

3.5.1 LSTM

In their study of computational models for empathy, Spring et al. (2019) determined that deep learning models yield best results. Specifically LSTM recurrent neural networks are advised. This model type is likely to work for empathy classification, as it has been used on similar text content analysis classification tasks previously, which is why the first model used in this study is an LSTM model.

The LSTM layer for which the LSTM model is named is a recurrent layer, which means that there is a recurrence looping over the data it processes. This recurrence is shown in the loop in figure 2a, which shows an example of recurrent neural networks in general. It can be thought of as a series of feed forward layers which each have an input in addition to the output of the previous layer as can be seen in figure 2b, with the major difference between the two being the shared weights in all steps in a recurrent neural network (RNN).

The input in each step makes RNN models very suitable for sequences of information with a start and an end, or data with a specific time associated with it. In each step of the recurrent loop, the output of the previous step is used in conjunction with a new section of data to produce a new output. These output are called ‘hidden states’ and are denoted in figure 2a and 2b by h_i . This enables each step in the recurrent layer to use information from the previous step. Because each output is affected by the previous step, each step is affected by *every* previous step, although effect size of any given step decreases with every step taken in the recurrent loop.

To be able to make use of information that was encountered more than a few steps ago, LSTMs were introduced (Hochreiter & Schmidhuber, 1997). The LSTM layer in an LSTM model has a so-called ‘cell state’, which is a mechanism which can store information and is separate from the direct transfer of information between steps. This cell state helps the layer retain information

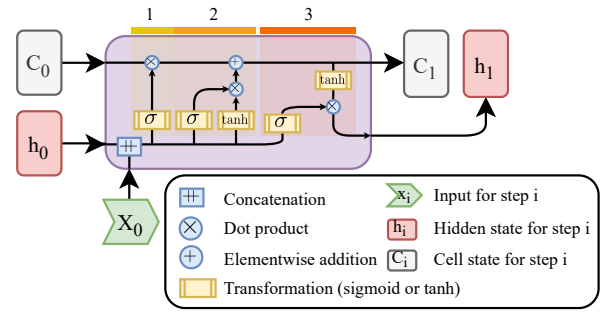


Figure 3: LSTM visualisation, see also Olah (2015).

from multiple previous steps and enables longer term dependencies to be resolved. Each step in the LSTM layer has two outputs, the current cell state and the layer output, feeding into the next step. In each step, the cell state is updated by adding a concatenation of the output of the previous step and the input for the current step to the input cell state vector.

Figure 3 shows the operations in each step in the LSTM layer. It is divided into three sections: deleting old values from cell state (1), inserting new values into cell state (2) and producing outputs (3). Figure 3 and the following section on LSTM models are adapted from Olah (2015).

Before section 1 in figure 3, the hidden state and the input for that step are concatenated. For the first step, the hidden state is a matrix with random initialisation weights and for every subsequent step it is the output of the previous step. This concatenation forms the input vector $[h_{i-1}X_i]$ for many of the operations in the layer.

In the first section of the LSTM layer, the values which are to be replaced in the cell state are deleted from the cell state. These values are determined by the input vector $[h_{i-1}X_i]$ scaled and offset by weight W_{del} and bias b_{del} and subsequently squashed by a softmax function, finally yielding f_{del} , the to be deleted values.

$$f_{del} = \sigma(W_{del} \cdot [h_{i-1}, X_i] + b_{del}) \quad (2)$$

The product of f_{del} and the previous cell state then yields the cell state with diminished weights C' , ready for new weights to be inserted.

$$C' = f_{del} \cdot C_{i-1} \quad (3)$$

In the second section of the LSTM layer, candidate values for the cell state are selected and inserted into C' . A softmax squashed scaled and offset input f_{ins} which determines the values to be updated (similar to section 1 where values were depleted) is multiplied with candidate values C_{can} , which has a hyperbolic tangent activation which is similarly scaled with a separate weight and bias.

$$f_{ins} = \sigma(W_{ins} \cdot [h_{i-1}, X_i] + b_{ins}) \quad (4)$$

$$C_{can} = \tanh(W_c [h_{i-1}, X_i] + b_c) \quad (5)$$

The cell state for the current step C_i is then calculated by adding

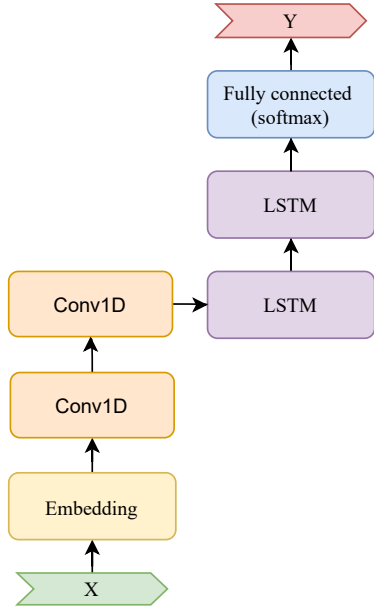


Figure 4: LSTM model as implemented by Khanpour et al. (2017).

the product of the insertion and candidate matrices (f_{ins} and C_{can}) to the prepared depleted cell state C' .

$$C_i = C' + (f_{ins} \cdot C_{can}) \quad (6)$$

In the third section of the LSTM layer, the new hidden state output is created, based on the combined inputs and the current cell state. A hyperbolic tangent activation function is used to squash the cell state, the output of which is multiplied with the input concatenation which is squashed with a sigmoid function.

$$h_i = \tanh(C_i) \cdot \sigma(W_{out} \cdot [h_{i-1}, X_i] + b_{out}) \quad (7)$$

This hidden state then serves as an input for the next step, along with the next item in the input sequence, usually the next dimension in the bag of words vector.

Reference LSTM models. The LSTM models developed by Khanpour et al. (2017) and Saeidi et al. (2016) serve as a basis for the LSTM model developed in this study. The models as described by Khanpour et al. (2017) and Saeidi et al. (2016) are also implemented in this study to serve as a comparison.

The model by Khanpour et al. (2017) uses two convolutional layers and two LSTM layers. The convolutional layers serve as trainable localized filters which can detect specific patterns in the data. The use of two convolutional layers enables the model to recognize patterns within the patterns detected by the first convolutional layer. As the data is one dimensional, one dimensional convolutional layers are used, with 64 filter channels with a size of three items. Figure 4 visualizes this model.

The model by Saeidi et al. (2016) is simpler and employs a bidirectional LSTM layer, which consist of two LSTM layers sequentially, with the second layer processing the input back to front. This model was adapted to give a single binary output but

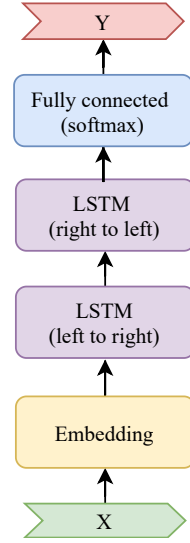


Figure 5: LSTM model as implemented by Saeidi et al. (2016).

retained the same activation. The Saeidi et al. model is visualized in figure 5.

Advantages and disadvantages. The biggest advantage of RNNs in general is their ability to take into account previous input when processing the current step. LSTMs increase this ability by enabling the model to retain information across many steps as the cell state is only partially updated every step. This context awareness allows these models to better model for example a negation, which can be very influential in the outcome of a classification task. It also enables the modelling of the meaning of a combination of words. This is useful in for example translation tasks, as sentences such as ‘I like to walk’ cannot be translated word for word in Dutch (‘ik wandel graag’). Here, it is useful to be able to combine the meanings of ‘like’ and ‘to’ into the single word ‘graag’.

This ability to use context does have its limitations. Because of the sequential nature of the recurrent models, only previously seen data can be used as context. There is no ability to alter previous steps with new information. In other words, the context awareness is one-sided. Bidirectional models attempt to circumvent this limitation by stacking two recurrent layers in a model, one processing the input from left to right, the other processing it from right to left. This enables the model as a whole to use two-sided context, but only one side at the time. This is the approach taken in the Saeidi et al. (2016) model.

Another disadvantage which is inherent to the dependency of each recurrent step on the previous step is the limited ability to parallelize. The steps in a recurrent layer have to be taken one by one, as they are dependant on the previous step.

3.5.2 Transformers

Since the study of Spring et al. (2019) was published, the Bidirectional Encoder Representations from Transformers (BERT) model has outperformed LSTM models in many natural language understanding tasks including sentiment classification tasks (Li et al., 2019; Sun et al., 2019). Since the empathy detection task requires a high level of natural language understanding, the BERT model is thought to be suited for this task as well. As the BERT model is

based on transformers, which in turn were the successors of the LSTM models, the transformer architecture is elaborated upon here as background for the BERT model.

Adaptions to the LSTM models. As mentioned, stacking LSTM layers proved useful in solving the bi-directionality problem, and this concept of stacking two LSTM layers was also applied in so-called encoder-decoder architectures. These models encode entire sentences as a representation vector, capturing the meaning of ‘I like to walk’ in a context vector and using a decoder LSTM to decode it in another language (Cho et al., 2014; Sutskever et al., 2014), hence these models encode input and decode into another feature space. Figure 6a shows a simple example of such an architecture.

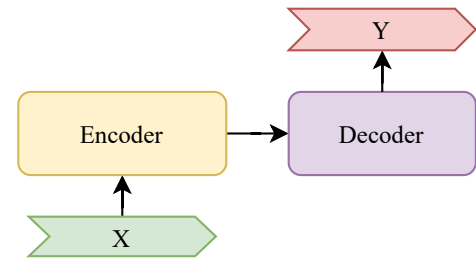
Another improvement was made by enabling models to select which cell state information in the LSTM layer is most relevant in the current step. This ability is called an attention mechanism. This attention mechanism formulates a query of what the model is modeling. The entire input sequence can then be compared to this query and appropriate focus can be put on specific parts of the input sequence. For example, in the machine translation task mentioned earlier where ‘I like to walk’ was translated to ‘ik wandel graag’, attention can be used to map the word ‘graag’ to both ‘like’ and ‘to’, even though they are not in the same location or even consist of the same amount of words. This ability to selectively use input features which is relevant at that point in the process proved very powerful and yielded good results in sequence to sequence models⁵.

The concept of attention proved so powerful in detecting which relations were relevant for the model, they were considered capable of capturing meaning without the use of LSTM layers (Vaswani et al., 2017). The encoder-decoder architecture with attention mechanisms and without LSTM layers is known as the transformer architecture.

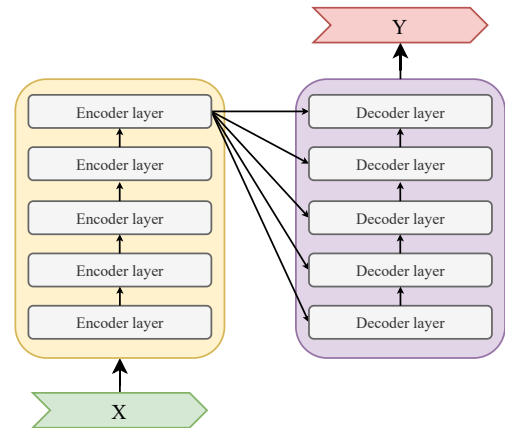
Transformer architecture. The encoder-decoder architecture which characterises transformers is visualised in figure 6a. Whereas the encoder and decoder in previous models consisted of LSTM layers, transformer models use feedforward neural networks in combination with attention mechanisms. Most transformer models use a stack of encoders and decoder layers instead of a single layer, as depicted in figure 6b. The encoder layers are arranged sequentially, each encoder processes the input from the previous layer. As the name implies, each encoding layer encodes its input into a different vector. These vectors are a representation of the meaning of the input. The attention mechanism can select which of the context features weigh in on the encoding of each feature. This yields context-dependent word embeddings, which are unlike traditional embedding layers in which embeddings are trained once and thereafter constant. A stack of encoders is able to represent the input in complex terms of meaning. This vector is then passed to every decoding layer, which decodes the meaning presented by the encoding stack into the output, which for example may be the same meaning in another language.

Figure 7 shows a detailed view of the encoding and decoding layers, and reveals that they themselves consist of layers. In each encoding layer in the encoding stack, self attention is used to incorporate related parts of the input sequence into the encoding of each word. For example, when encoding the word ‘it’ in the sentence ‘the tea is cold because it is iced tea’, the words ‘the’

⁵As this study is not concerned with sequence to sequence models, the reader is referred to Bahdanau et al. (2014) and Luong et al. (2015) for elaborations on several implementation strategies for attention in such models. The attention mechanism used in the BERT model as used in this study is explained in section 3.5.3.



(a) Basic transformer architecture.



(b) Transformer architecture layers.

Figure 6: Architecture of most transformer models. Figures adapted from Alammari (2018).

and ‘tea’ are relevant for the meaning of ‘it’ and are therefore included in the encoding for the word. Other words such as ‘because’ do not contribute to the meaning of the word ‘it’ and are not included in the encoding. Multiple attention ‘heads’ will process the input simultaneously, yielding a concatenation of different self attention vectors. These different heads enable the model to attend to different things simultaneously. A feedforward neural network is then used to combine these different attention heads into one encoded output and reshape the output to fit the next encoder layer.

The decoder layers in the decoder stack use a similar process to form an output sequence. A self-attention layer models the relation of each word with regards to each other word. An encoder-decoder attention layer maps relations between the encoder output and the input from the previous decoder layer. Eventually, the decoder stack produces an output which can be used for classification or sequence modelling, usually in the form of a distribution over a vocabulary.

As claimed in ‘Attention is all you need’ (Vaswani et al., 2017), the attention mechanism is capable of representing meaning well on its own. The many attention layers in the transformer model leverage this power to produce models which perform well on various language understanding tasks.

Transformer models are still sequential (in the sense that they cannot process one sequence fully parallel) because the decoder is dependant on the previous output. For example, in a translation task each word is formed by taking the encoded embedding, positional embedding and previous output (for first word this is sentence start token).

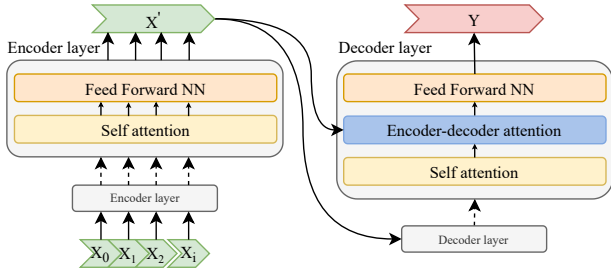


Figure 7: Detail view of encoder and decoder layers in transformer architectures.

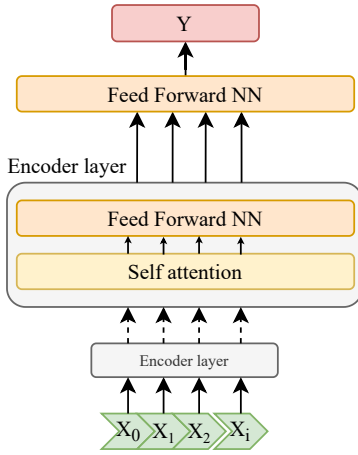


Figure 8: BERT resembles the encoder stack found in transformers (see figures 6 and 7, left side).

Since the LSTM layers were replaced with attention and feed-forward layers, more parts of the model have been stripped in favour of attention. In the Bidirectional Encoder from Transformers (BERT, presented by Google (Devlin et al., 2019)), the entire decoder stack is absent, and the model resembles the encoder stack of a transformer model.

3.5.3 BERT

The BERT model is currently one of the best performing models on natural language understanding, as shown in various benchmarks where BERT derived models dominate the leaderboards⁶. The BERT model is built up like the encoder stack from a transformer, but without decoder. The encoder stack counts 12 layers instead of the usual 6 or so found in transformers. Since the encoding layers produce a context vector and not an interpretable output sequence, a feedforward layer is usually used to perform classification or output generation based on the context vector.

BERT is capable of high performance natural language understanding because it is capable of representing complex patterns in its weights and because it is trained on a very large dataset (over one billion words, consisting of a corpus of books and wikipedia). To be able to train this complex model without the need to annotate all of the data, training tasks were constructed using data which was already in the texts. These training tasks need to be hard enough to require a large degree of understanding of the

language data in order to train the representations to encode meaning properly. Two training tasks were employed in parallel while training the BERT model: masked language modelling and next sentence prediction.

In the masked language model task, a small percentage (between 10 and 15%) of the wordpiece tokens which are input into the model are marked to be masked. The model then predicts each missing token based on the encoded context vector arising from the masked token. The model has to use words around the missing word to encode the meaning of the missing word into the output vector. To prevent the model from *solely* using context to determine what each word means, not every word is actually masked. 80% of the words marked to be masked are replaced with the special [MASK] token, which indicates that there is a token missing. Of the remaining marked tokens, half is replaced with a random other token and half is kept original. This ensures that the model is also trained to take the word itself into account when determining its meaning.

In next sentence prediction, the model is given two pieces of text, A and B, and is tasked with determining whether text B follows text A directly. This helps the model train to capture relationships between texts and capture meaning across features. Texts A and B can be any length but are sampled such that the combined length is smaller than the total input size which is 512 embedding features long. Text B follows text A 50% of the time.

Bert input. The BERT model is trained on wordpiece word features (figure 9, X_i , see also section 3.1), additional features can be added after the [SEP] token (figure 9, F_i). The wordpiece features are presented in the order of occurrence in the text, so no bag of words or TF-IDF text representation is used. Despite the being in original order, the model has no concept of order or sequentiality, which is a consequence of the full bidirectionality of the model. The weights which are used are dependent on the input, not on the order of the words. To provide the model with information on which word is placed where, this information needs to be encoded into the input. This is done by adding a positional embedding vector to the input before the first layer. This positional embedding vector is trained with the model, unlike in for example transformer models in which it is hard-coded. This positional embedding is indicated as the P_i sequence in figure 9.

To facilitate the training tasks, a sequence embedding is also added to the input, which divides the input into two sequences, allowing the model to differentiate texts A and B for the next sentence prediction training task, S_A and S_B in figure 9. In the token embeddings, a [SEP] token is used to separate text A and B. For the same task, a dedicated spot in the embedding is reserved as a classification output. This takes index 0 in the token embeddings and is represented by the [CLS] token. Padding appear as [PAD] in the token embedding. In this study, sequence B is used to encode features not taken from the text, such as user title, thread topic and thread tags.

Attention in BERT. As was mentioned in section 3.5.2, attention mechanisms allow the model to prioritize part of the input over another, depending on what input data is important. In the BERT model, attention plays a central role. This section elaborates on how some data is prioritized and how the model selects what is important.

In the self-attention terminology for the BERT model, a comparison is made with retrieving a value from a database. As such, there is a 'query', a 'key' and a 'value' matrix for each feature.

⁶GLUE: <https://gluebenchmark.com/leaderboard>,
MultiNLI: <https://paperswithcode.com/sota/natural-language-inference-on-multinli>,
SQuAD: <https://rajpurkar.github.io/SQuAD-explorer/>

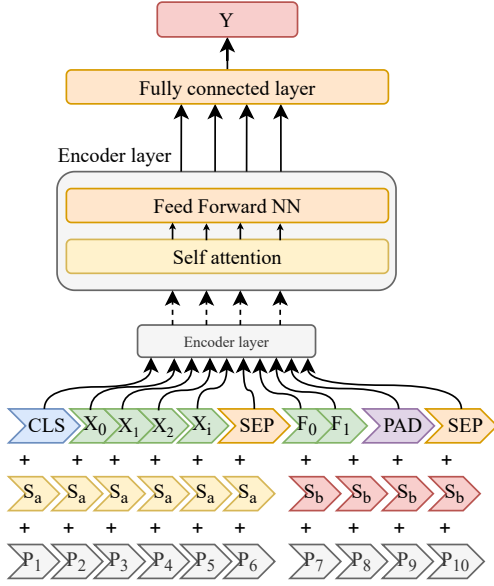


Figure 9: BERT model inputs, input tokens vector with classification (CLS), wordpiece tokens X_i , separation tokens (SEP), additional feature tokens F_i and padding token (PAD), sentence part vector with parts A and B and positional embeddings vector consisting of P_i dimensions.

These matrices have different roles in determining where attention is focused. The query, being a ‘request for information’ can be seen as a representation of what needs to be known for that feature. The key, with which the query is compared, represents the knowledge that a feature can add to the encoding of another feature. The value is what is eventually used to add meaning to another feature, and the extent with which this is done is determined by how well the query and key match.

To illustrate how this works follows an example in which the word ‘Ernie’ in the small sentence ‘Ernie smiles’ is processed. This example is visualized in figure 10. The query, key and value vectors for ‘Ernie’ are q_0 , k_0 and v_0 , and the vectors for ‘smiles’ are q_1 , k_1 and v_1 . These vectors are the product of the embedding with the weights W_i^Q , W_i^K , and W_i^V for the query, key and value respectively.

When encoding word 0 (Ernie), the query vector q_0 is multiplied by the key vector k_0 to produce score S_0 . Then, query vector q_0 is multiplied by key vector k_1 to produce score S_1 , indicating how much word 1 (‘smiles’) should contribute to the encoding of word 0 (‘Ernie’). If all scores are calculated, they are divided by the square root of the number of dimensions in the key vectors, which is 64 in the model used in this study. A softmax function maps these scores to values between 0 and 1 such that the sum of the scores is 1.

These squashed scores are used to scale the amount with which each word contributes to the word which is currently encoded. The squashed score S_0 is multiplied with value vector v_0 to produce Z_0 and squashed score S_1 is multiplied with value vector v_1 to produce Z_1 . These weighted value scores are summed to produce the weighted value Z for word 0. This value is a combination of meanings from different words, the proportions of which are determined by how well the query and key vector correspond.

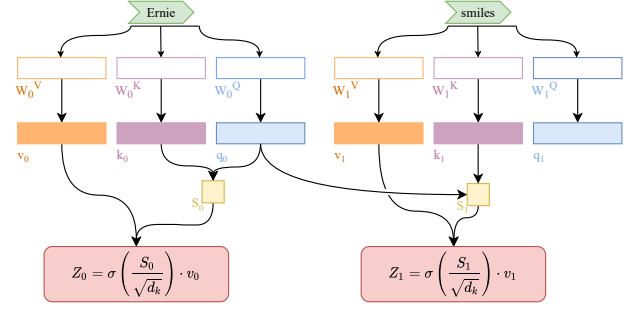


Figure 10: Attention mechanism in BERT, showing one attention head processing word 0.

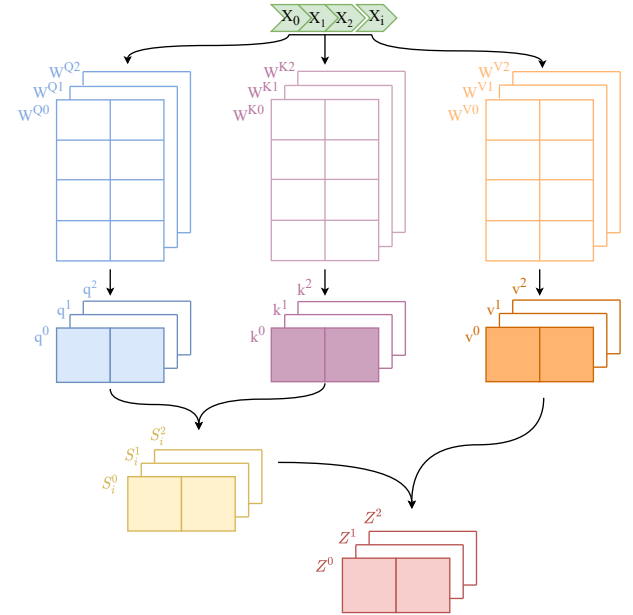


Figure 11: Multihead attention in BERT

Since both the query and key vector are a representation of meaning for that word, they are usually quite similar. This makes the calculated score S large for that word, which ensures that the word itself contributes a large amount to the encoding. This makes sense, as the meaning of the word itself should have a large impact on the embedding.

The attention mechanism does not consist of only one set of query, key and value matrices but of 12 sets, which are called ‘heads’. Each head has its own W^Q , W^K and W^V weight matrices and is able to learn different things to attend to. It might seem counterintuitive to have different value matrices for the different heads as well as different query and key matrices but these value matrices represent what a word means in the context of that attention head.

Every word in the sentence is trained in parallel, as are all heads. Figure 11 visualises this. Each of the vectors shown in figure 10 are a row in the matrices in figure 11. The attention heads produce 12 weighted value vectors for each word, each one a sum of the values of all other words in the context. To reshape the 12 vectors into one vector which can be processed by the next layer, the output of each head is concatenated so that each word is represented by a vector of 768 (12 times 64)

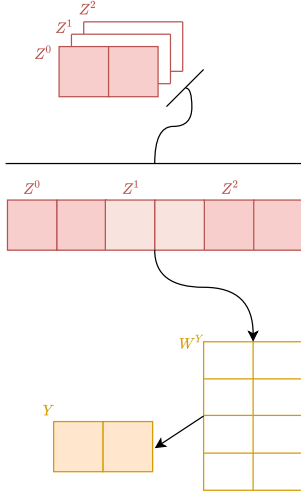


Figure 12: Multihead weighted value matrices to output

dimensions. This vector is projected to a vector of length 64 through a weight matrix which is also trained with the model. This matrix combines the attention head outputs to a final output of the attention layer. Figure 12 shows the concatenation and output producing step.

3.6 Model comparison methods

The BERT model has been proven to work well in its pretrained form in English, but might require more data to fine-tune than is available. Additionally, the pretrained Dutch model might not transfer well to the Kindertelefoon data. Because of these reasons, both the LSTM model and the BERT model are implemented.

Many scoring formulas for assessing the performance of machine learning models exist, the most commonly used being the accuracy score as shown in equation 8. The accuracy score is the proportion correctly classified documents with regards to all documents. This score does not take into account skewed distributions of the classes, as a classifier which predicts only one class regardless of the input would perform well if the vast majority of the cases do in fact belong to that class. As the majority of documents in the Kindertelefoon data are not an expression of empathy, the accuracy metric easily leads to a misleading performance score.

$$\frac{C_1^T + C_2^T}{C_1^T + C_2^T + C_1^F + C_2^F} \quad (8)$$

Another often used performance measure is the area under the receiver operating characteristic (ROC) curve. The ROC curve is a plot of true class predictions versus false class predictions for binary classification. The area under the curve is an indication of how well the two classes can be separated by the classifier. Like the accuracy score, the area under the ROC curve shows a misleading performance if the distribution is skewed, as the number of false class predictions is very low if the majority of the data is one class.

	Class 1	Class 2
Pred. Class 1	C_1^T	C_1^F
Pred. Class 2	C_2^F	C_2^T

Table 1: Confusion matrix

Confusion matrices are used to obtain more information about the class predictions. Figure 1 shows a confusion matrix for two classes, class 1 and class 2, in which the predicted classes and actual classes are laid out. If a confusion matrix shows many correctly predicted documents for both classes (marked by superscript T) and few falsely labeled documents (marked by superscript F), the classifier performs well. To summarize this matrix, a number of metrics exist.

The precision metric quantifies the fraction of correctly labeled documents which are predicted to belong to one class. The recall is a measure of the fraction of correctly labeled documents which actually belong to one class. They are useful indicators of performance, but both tell only part of the story how well a model performs. The precision remains unchanged regardless of how many documents are falsely labeled. The recall metric is only sensitive to the performance on one class, not the other.

A measure which takes both precision and recall is the *F1* score, which is the harmonic mean of the precision and recall.

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2 \cdot C_1^T}{2 \cdot C_1^T + C_1^F + C_2^F} \quad (9)$$

It is notable that the true class 2 (the true negative) prediction is missing from the *F1* formula. Because of this, it is still sensitive to misleading performance scores when the data is heavily skewed (Chicco & Jurman, 2020).

The Matthews Correlation Coefficient (MCC) is a performance measure which includes all four quadrants of the confusion matrix. The score, shown in equation 10, is 1 if all classes are perfectly predicted, -1 if no classes were predicted correctly and 0 for chance level prediction. As the MCC equally weights performance for both classes and uses proportions of both classes as a basis for performance, it is not sensitive to imbalanced data.

$$MCC = \frac{C_1^T \cdot C_2^T - C_1^F \cdot C_2^F}{\sqrt{(C_1^T + C_1^F) \cdot (C_1^T + C_2^F) \cdot (C_2^T + C_1^F) \cdot (C_2^T + C_2^F)}} \quad (10)$$

4 Dataset background

The child help volunteer organisation De Kindertelefoon has provided a phone line that can be called since their establishment in 1979 and have offered an online live chat since 2008 and a public forum since 2012. On all channels, children are able to talk with adult volunteers though on the forum, they can also talk to peers. Any subject is allowed, but on the forum there are rules regarding privacy and explicitness. The most common subjects that children talk about are: relationships, sexuality, bullying, home and family, bodily development, and spare time activities.

The forum of the Kindertelefoon organisation allows children between ages 12 and 18 to post questions, issues, rants and advice of any sort, about any subject on the forum. These posts are sorted into fifteen topics, the distribution of which can be seen in figure 13.

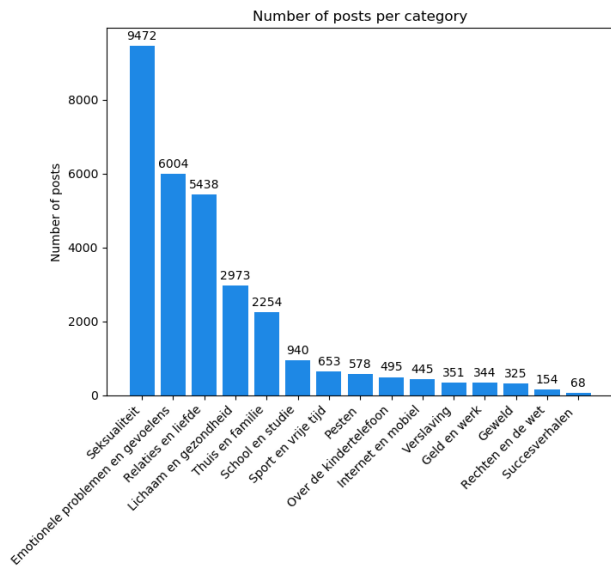


Figure 13: Distribution of threads per topic.

The posts on the forum can be categorized into three types. There are generic non-personal questions, which are often broadly formulated and often take the form of a question list (one can be seen in figure 16a). They allow users to compare themselves with peers by comparing answers to the question list with their own. These threads often do not spark a discussion, but consist only of filled in question lists.

Personal questions, which are often highly specific towards the poster’s situation, are often responded to with advice by peers and/or Kindertelefoon volunteers. These questions are concerned with how to deal with certain specific situations in a user’s life, whether their concern is valid, and what others would do.

Lastly there are expressions of feelings, or vents. Like the personal question posts, expressions of feelings or vents are specific to the context of the poster and are answered with compassion and advice, again by either peer or volunteer. These types of posts may not explicitly ask how to deal with situations but rather express frustration, anger, sorrow, confusion or worry.

Identifying posts which call for empathy may benefit the Kindertelefoon forum volunteers, as these posts likely require their attention. The models developed in this study can be applied to alert volunteers to certain messages or help them prioritize message handling. An empathy detection algorithm can also assist non-human response agents in building up an appropriate response to a post in a later stage or provide a nudge for peers or volunteers to write a certain type of response which is found to be appropriate. In general, the language models developed in this study can be applied to detect other patterns in the data which might be important in managing the forum.

4.1 Kindertelefoon forum website functionality

Users have to indicate their age when they make an account on the forum. This account is blocked when the user turns 18, but is not deleted. Instead, the posts made by the account are labeled ‘Anonymous’ and the user overview page is cleared. Threads to which has not posted for over six years are deleted. They may

also be deleted if they violate forum rules or compromise the identity of a user.

Every user has a usertitle associated with them. This usertitle is based on the number of posts that child users have posted and also differentiates the child users from Kindertelefoon official volunteers. These titles range from ‘Just new’ to ‘Familiar’ to ‘Star’ and ‘Hero’.

Forum users can mention or cite other users in their posts. These are often users who responded previously in the same thread although this is not always the case.

The forum has an equivalent of a ‘like’ button, which takes the shape of a four-leaf clover. Users can give a post on a forum a four-leaf clover in order to express that the post moved them emotionally. The forum states that the four-leaf clover is intended to express support and that the four leaves of the clover represent hope, faith, love and luck (Kindertelefoon, 2018).

4.2 Data descriptives

In total, 221707 messages were downloaded which cover 30494 threads, which means that on average threads contained just over 7 messages. For the annotation dataset, 1500 threads were drawn from the full dataset such that the proportion of messages per topic is equal to the full dataset. Of these 1500 threads, 18 threads contained more than 100 messages. These threads were not included in the annotation dataset. The remaining 1482 threads contained 9776 total messages. The annotators which remained after removal of poor performers annotated a total 6651 messages in 1034 threads.

The total amount of spelling errors as evaluated by the opentaal dictionary (Opentaal, 2020) was 992.028, which is on average 4.5 mistakes per message. It is noteworthy that many of these not true mistakes, as the used dictionary did not contain numbers. Numbers between 2 and 10, which are most likely counts not written out and numbers between 12 and 16, which are most likely ages, were most used. The total number of times an integer was featured in the texts was 297.079⁷. Usernames from the forum which were mentioned in the messages made up 95.544 of the total out of vocabulary (OOV) words for the dictionary comparison. This leaves 599.419 mistakes without usernames and without numbers. The fifty most common of these words are laid out in table 2. In this table, integers are not included.

Most common OOV words are contractions or abbreviations, such as ‘ofzo’ (meaning *something like that*) and ‘mn’ (meaning *my*). English words such as ‘sex’⁸, ‘nope’ and ‘twin’ as well as English abbreviations such as ‘idk’ (meaning *I don’t know*) are featured as well.

Due to the wordpiece tokenization, the final count of unknown words which is processed by the BERT model is much lower than the 992.028 spelling mistakes. In the entire dataset, the [UNK] token was used 459.883 times, out of the total of 24.493.200 tokens which comprise the whole dataset. These tokens were made up of 13.997 unique words, the fifty most common of which are presented in table 3.

Most of the words which were not parsable by the BERT tokenizer were individual letters. The most common one, ‘n’, is a result of the incorrect tokenization of the Dutch contractions ‘zo’n’ into ‘zo’ and ‘n’ and ‘m’n’ into ‘m’ and ‘n’, which also explains the ‘m’ on number 5. The individual letter ‘t’ is a result of the abbreviation of ‘het’ into the single letter ‘t’. The letter ‘s’

⁷Of which 14 are also author names

⁸Although this could also be a misspelling of the Dutch translation *seks*.

#	OOV word	Amount	#	OOV word	Amount
1	ofzo	12819	26	Hoevaak	1612
2	mn	12765	27	oké	1587
3	gr	10529	28	nie	1553
4	gwn	8304	29	ok	1537
5	enzo	7455	30	dr	1536
6	sex	7292	31	nope	1500
7	xx	6609	32	hoevaak	1486
8	etc	6113	33	ookal	1445
9	zoja	5684	34	ofz	1355
10	ff	5061	35	hij/zij	1338
11	hey	4765	36	Oke	1323
12	bijv	2814	37	-Dr8gon99	1290
13	Heyy	2646	38	Mn	1288
14	Xxx	2554	39	Nederland	1279
15	mastruberen	2513	40	xD	1271
16	oke	2422	41	twin	1256
17	Heey	2405	42	crush	1205
18	xxx	2358	43	it	1180
19	idd	2290	44	luca	1169
20	idk	2265	45	2x	1159
21	you	2171	46	da	1109
22	jongen/meisje	2060	47	zegmaar	1094
23	sws	1927	48	jongens/meisjes	1089
24	zn	1676	49	gezezt	1062
25	Nope	1649	50	forumregels	1046

Table 2: Fifty most common spelling mistakes according to the Opentaal (2020) dictionary.

#	OOV word	Amount	#	OOV word	Amount
1	n	20882	26	sws	1927
2	t	14060	27	anaal	1627
3	mn	12765	28	cup	1541
4	@	12789	29	ok	1537
5	m	9761	30	•	1322
6	gwn	8304	31	p	1309
7	x	7899	32	u	1299
8	xx	6608	33	xD	1271
9	s	5708	34	twin	1256
10	ff	5061	35	\$	1199
11	k	4624	36	it	1180
12	ie	3886	37	c	1169
13	Xx	2700	38	se	1093
14	%	2574	39	i	1073
15	Xxx	2554	40	icon	1070
16	oke	2422	41	nvt	1007
17	xxx	2358	42	o	1001
18	eet	2337	43	iig	967
19	idd	2290	44	sixpack	912
20	idk	2265	45	tieten	891
21	X	2195	46	überhaupt	873
22	Cupmaat	2174	47	XD	858
23	a	2152	48	\\	848
24	=	2052	49	ivm	847
25	neit	2003	50	oh	796

Table 3: Fifty most common words which are not parsable by the BERT tokenizer.

	Value count	
	0	1
Call for empathy	5904	1154
Is empathy	6723	335
Question	4908	2150
Answer	3882	3176
Call to action	5764	1294

Table 4: Distribution of labels for most important features.

comes from the possessive suffix ‘s’ which is tokenized separately from the word of which it signifies possession.

Some real words are not parsed, notably ‘eet’ (*eat*), ‘oke’ (*okay*) and ‘cupmaat’ (*cup size*). There are also abbreviations such as ‘nvt’ (*not applicable*) and ‘ivm’ (*in relation to*) and slang abbreviations such as ‘mn’ (*my*) and ‘ff’ (*for a moment/only*), English words such as ‘icon’ and ‘sixpack’ and abbreviations such as ‘idk’ (*I don’t know*). There are many variations on x’s which are used to sign off messages throughout the unparsable list.

There were some none alphanumeric characters which failed to be parsed. Number 30 in table 3 is a bullet from bulletpoint lists. There were also some emoji’s, though they are not used widely on the Kindertelefoon forum, none of which could be parsed. The *Face with Tears of Joy*, *Smiling Face With Open Mouth* and *Cold Sweat*, and *Thinking Face* are the most commonly used emoji’s.

In total, there were 169 unique annotators who contributed at least one annotation. At least 110 come from Mechanical Turk, as these annotators can be traced to the MTurk platform either through their website username or through email contact. There were 29 annotators who participated through the Sona platform, which leaves 30 participants which have been recruited through social media and direct messages.

A skewed distribution of labels was expected, which drove the choice for the evaluation methods. The final distributions of labels for the most important features is shown in table 4, which indicates a skewed distribution as expected, especially for the ‘is empathy’ label.

The most commonly labeled antecedent was the first post of a given thread. In total, the antecedent of 2016 of the 6651 messages were labeled as the first post. A total of 1130 messages had the second post as antecedent, 560 messaged the third and 445 messages the fourth. The thirty most frequently encountered antecedents are listed in appendix F, as are the predictions for the reply relation resolution algorithm, which are very similar in count to the antecedent labels.

5 Methods

The main goal of this study is the development of a classification algorithm for identifying texts which contain expressions of empathy in a corpus of Kindertelefoon forum messages. To establish the definition of what constitutes empathy in the context of this study, related works were considered (section 2) and interviews were conducted with psychologists, which is described later, in section 5.1.

As was described in section 2.4, post relation information is necessary to determine a call for empathy through posts containing an empathetic response. As this is not encoded naturally

in the Kindertelefoon data, an algorithm was developed to extract relation information from content and meta-features. The development of this algorithm is described in section 5.2.

Section 5.3.1 describes the features which will be used in the language models, which are based on the empathy definition established in related works and interviews and on the features available from the Kindertelefoon forum web page.

The language models used for both the empathy and call for empathy classifiers are described in section 5.4, as are the model comparison methods.

5.1 Interview

To get a better understanding of what nuances exist in the specific demographic which is targeted in this study, two interviews were conducted⁹. The two interviewees were both experts in the field of technology mediated narratives and emotions. The interviews covered the definition of empathy in the context of online help fora for teenagers. The (Dutch) interview questions can be found in appendix A.

The aim of the interviews is to get an understanding of which constructs elaborated upon in section 2.1.1 are specifically relevant within the scope of this project. As mentioned in Batson (2009), some concepts are closely related while others are more distinct. To match concepts of empathy as closely as possible to context of the Kindertelefoon forum, some of these concepts are merged using the input from the interviews.

The interviewees were asked to list what in their mind constituted empathy. These constituents need not all be present simultaneously but should be an important part of an empathetic response or experience. The interviewees were asked to relate the relations depicted in figure 14 to the constituent. This was not a dichotomous question, but rather an indication of what role the constituents played in each relation. This enables small differences in empathy constituents between adult-child, adult-adult and child-child expressions of empathy to be captured. These nuances are important to define as most literature does not capture these distinctions.

In order to group the concepts laid out in section 2.1.1, the psychologists were asked to group the different concepts in a number of groups as small as possible while maintaining a distinction between them. This enabled a grounded grouping of concepts to be used later in annotating the data.

Another goal of the interviews was to determine different strategies for identifying empathy in text. For this, interviewees were asked what components of empathy are expressed in a textual way and how they would go about recognizing an empathetic response. For the detection of a call for empathy, similar questions were asked. The main focus here was on what components of a post make an empathetic expression appropriate. For both the empathy recognition questions and the call for empathy recognition questions, interviewees were asked whether there is a difference between adult-adult, adult-child and child-child expression of empathy.

⁹A focusgroup session with Kindertelefoon employees was prepared as a practise-based grounding alongside the psychologists interviews. The session was meant to give a better understanding of the types of questions children ask, the nature of the situations that are encountered and the data that was processed as well as insight on the differences between messages on the different media that the Kindertelefoon employs (chat, phone and public forum) and the interaction between them. The focus group also served to validate assumptions made about the data, the concept of empathy and the selection of posts to which to reply by the volunteers. The focusgroup session could not take place because the Kindertelefoon was unavailable at the time of study.

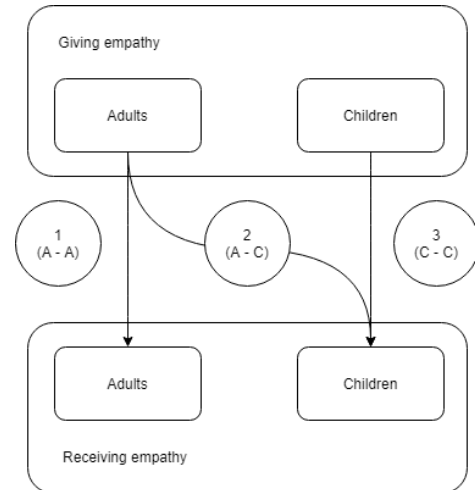


Figure 14: Visualisation of empathizer-target relations of adults and children on the forum.

In the open question part of the interview, the indicators of empathy that were mentioned most were *recognizing feelings another may have* and *providing feedback* about that recognition, which fits in with the first definition of empathy as described in section 2.1.1.

Both interviewees felt like there was an action component missing from the empathy definition. In many posts on the forum which answer help-seeking questions, a call to action is made. This is often preceded (or at least accompanied) by an empathetic response but is not in itself a part of the empathetic response. One interviewee proposed ‘compassion’ as a concept to use to detect which messages should be responded to. The concept of compassion includes a recognition another person’s emotions and may include emotional resonance, which can be seen as empathetic, but also include a call to action to alleviate suffering (Strauss et al., 2016). This description fit the responses seen on the forum well, but other aspects of compassion, such as the focus on suffering and the recognition of the universality of human suffering do not stroke with the content on the forum. Many questions and posts on the forum, including ones which call for empathetic responses, are not concerned with suffering. Such posts include questions about feelings, going through puberty or generally any question which is too embarrassing for the users to ask real life friends or parents. Since the ‘call to action’ is a significant part of the help which is offered on the forum, it was considered as a feature in this study, but since the concept of compassion has a poor fit in other aspects, the core concept remained empathy.

In comparing what empathy means for adolescents with how it is described in academic literature, the interviewees indicated that both in the call for empathy as well as the giving empathy part of the interaction, adolescents will respond differently than adults. In the call for empathy, adolescents may not immediately or clearly indicate what is really going on. Although circumstantial descriptions are not absent in adults, they are expected to be more prevalent in adolescents. For this reason, follow-up questions to find out what is actually happening or what the message actually means may be more important in this context than it is for help seeking messages for adults. With regards to providing empathetic responses, adolescents may not be as skilled as adults

in registering what another person is going through and providing appropriate feedback indicating this. However, no suggestion was made that specific components of empathy would be more or less appropriate for this age bracket.

In the part of the interview in which the constituents of empathy are reviewed and grouped, one interviewee noted that the constructs form a scale of a small to a large amount of empathy as opposed to discrete components which exist in parallel in terms of amount of empathy. The combination of constituents four and five (empathy as imagining how another is thinking and feeling & empathy as literally perspectivising) was the only grouping found in both interviews. Constituents seven and eight (empathy as feeling distress because of another person's malaise & empathy as feeling for another person's suffering) were found to be close as they are both concerned with the empathizer feeling distress, but were found to be different after all because of the reason behind the feelings of distress.

Constituent two (empathy as physical mimicry) was not found relevant by interviewees, as many of the triggers of the physical mimicry are absent in the online context of the forum.

5.2 Relations between posts

As mentioned in section 2.4, several similar studies list several post relation types. These include question-answer pairs, examples, similarity, temporal sequence, elaboration, (dis)agreement and courtesy relations. Some of these relationships are easier to detect than others. The question-answer pairs, temporal sequence and similarity relations have the most literature to support a grounded method. Hence, these are the relationship types that will be covered in this project.

Methods for finding these relationships are described in section 2.4 and include simple temporal relation detection, similarity scores (Y. Wang et al., 2008), unsupervised learning (Cong et al., 2008; Shrestha & McKeown, 2004), supervised learning (H. Wang et al., 2011), heuristics and non-content features (Aumayr et al., 2011; El-Assady et al., 2018; Kim et al., 2010) and several combinations of these methods. In this study, a combination of meta features, similarity, supervised learning and heuristics were used.

These methods are distinct, and are represented as a complex decision tree. The tree is complex in the sense that each branch has additional conditions which need to be fulfilled and in the sense that multiple paths lead to the same end node. The nodes in the tree are placed in order of confidence, the strongest indications of a response relation between posts is checked first, the weakest last. Every subsection in this section of the report represents a main node in the tree, and as such the order of the subsections represents the order in which a post relation is evaluated. A high-level representation of the dependency tree can be seen in figure 15. In an effort to contain the complexity of the dependencies, each post is assumed to be a response to exactly one other post. This is considered overly simplistic by some (El-Assady et al., 2018; Y. Wang et al., 2008; Wolf & Gibson, 2005), but is necessary because of the limited scope of this project.

5.2.1 List type post

In the first node in the tree, the post type was identified. All responses in the list type threads were assumed to be a response to the original post. Such threads start with a series of questions and are answered by several users, often repeating the questions when adding answers. An example of such a list type post can be seen in figure 16. Both 16b, and 16c are responses to a question list.

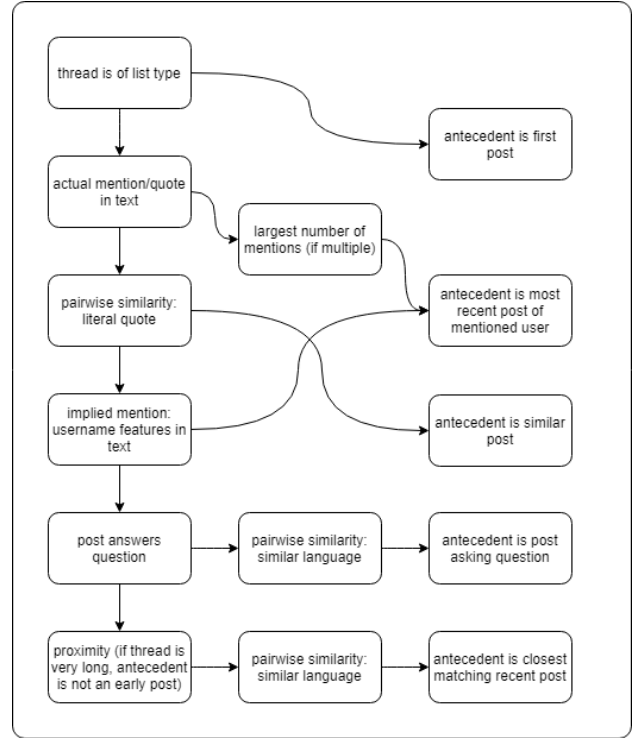


Figure 15: Dependency builder architecture.

This figure shows how similar the responses are to the original post, as even the question at the top and the closing statement at the bottom are copied over to the responses in this example. This is an example of the dataset-specific features which mentioned by El-Assady et al. (2018).

5.2.2 Literal mention and quote

Mentions and quotes made with the mention or quote mechanisms on the website were checked next. If a post was quoted, the quoted post was looked up and marked as the antecedent. If a user was mentioned, the last post in that thread by the mentioned user was marked as the antecedent. If multiple users were mentioned, the most frequently mentioned user was picked. If this was a tie, the first mentioned user of the tied users was picked.

5.2.3 High similarity/implied quote

Direct quotes of previous posts were detected through analysing the similarity between the posts. A very high similarity between posts can only be caused by a citation from a previous post, even if the citation was not made through the built-in quotation function on the forum. If a very high similarity between a post and a previous post is detected, the previous post is assumed to be quoted and marked as the antecedent of the post being checked at that time. This does not hold for the 'list' type post, as can be seen in figure 16. In list type posts, responses have high similarity among each other and may have higher similarity to each other than to the post they reply to. For this reason, the lists type posts were identified before the high similarity/implied quote step.

The method of calculating similarity scores between posts was the same as used in Y. Wang et al. (2008), who used the TF-IDF similarity score to detect post structure in a newsgroup type of online environment. Firstly, a vocabulary of all words

Heyy ik heb een paar vraagjes voor jullie ;)
 1 hoe oud ben je
 2 jongen/meisje
 3 heb je een religie zoja welke
 4 ben je wel eens onzeker zoja waarover
 5 wat is je lengte
 6 naar welke soort muziek luister je
 ik hoop dat jullie hem willen beantwoorden!!

(a) Example of a 'list' type post, asking questions about the lives of peers.

Heyy ik heb een paar vraagjes voor jullie ;)
 1 hoe oud ben je 14
 2 jongen/meisje JONGEN
 3 heb je een religie zoja welke NIET ECHT
 4 ben je wel eens onzeker zoja waarover ME BUIKJE
 5 wat is je lengte 170 CM
 6 naar welke soort muziek luister je POP
 ik hoop dat jullie hem willen beantwoorden!!

(b) One response to the question list.

Heyy ik heb een paar vraagjes voor jullie ;)
 1 hoe oud ben je 14
 2 jongen/meisje jongen
 3 heb je een religie zoja welke christelijk
 4 ben je wel eens onzeker zoja waarover niet echt maar misschien mn gewicht
 5 wat is je lengte 167
 6 naar welke soort muziek luister je top 40 meestal dsu eog pop
 ik hoop dat jullie hem willen beantwoorden!!

(c) Another response to the same list, very similar to the post in figure 16b.

Figure 16: Two responses to a list type question post. Neither are the original post, both have copied the original post and added answers to the questions. These responses show incorrect high similarity antecedent assumption for list type posts, as these posts are more similar to each other than to the post they respond to.

in the thread was made. The TF-IDF score was calculated for every word in the vocabulary for each post in the thread (see equation 11). This resulted in a weighted encoding for each document, weighted by the term frequency in each document and the inverse document frequency of each word. This weighting was applied to lower the distinguishing ability of words which occur frequently in the entire thread. The post similarity score was finally calculated by multiplying the thread TF-IDF matrix with a transposed TF-IDF matrix, which yielded the similarity matrix. Table 5 shows an example of such a similarity matrix, in which post 4 quotes post 1, which is indicated by the high similarity between the posts.

	1	2	3	4
1	1			
2	0,21	1		
3	0,18	0,02	1	
4	0,91	0,05	0,14	1

Table 5: Example of a similarity matrix indicating a direct quote of post 1 in post 4.

Different implementations of the TF-IDF formula exist, the following ($k = 1$ smoothed) formula was used.

$$TFIDF_{t,d} = tf_{t,d} \times \log \left(\frac{1 + N}{1 + df_t} \right) + 1 \quad (11)$$

in which $tf_{t,d}$ is the raw term count in a document, N is the total number of documents and df_t the number of documents in which the term occurs.

5.2.4 Implied mention

Not all mentions of users were made using the mentioning mechanism on the forum. If a user was mentioned by having their username typed out, it does not show up as a mention as processed in section 5.2.2. Because of this, posts were searched for usernames of users which had previously posted in the same thread. As was done for the proper user mentions, the last previous post for the mentioned user was selected as antecedent for the post in which that user was mentioned. The implied mention was considered a less confident indication of post relation than the implied quote (the previous node in the tree) because the implied quote referred to a specific post, while the implied mention referred only to a user and might have referred to any post by that user.

5.2.5 Question/answer relationship

As mentioned before, Xi et al. (2004) have proposed five types of relationships which posts on a forum may have, among which are question and answer relationships. To detect question-answer relationships between posts, firstly the presence of questions was detected. Then, all posts which were considered answers to the question had the question post marked as antecedent if they had not already been assigned an antecedent in one of the previous steps.

Cong et al. (2008) and Shrestha and McKeown (2004) have proposed new methods to improve upon simple keyword detection for question detection (see section 2) and have obtained significantly improved results over a simple keyword detection method. For the Kindertelefoon dataset, the same results were not guaranteed since the language used is simpler and contains more spelling and grammar errors, slang and contamination from English than the travel guide forum datasets that were used in Cong et al. (2008) Figure 17 shows an example of a post with such spelling errors. The presence of errors, slang and contamination hinders the use of n-gram model because many n-grams are unique. This makes the representation more sparse, which hinders performance. On the other hand, the smaller vocabulary of the forum post authors may reduce sparsity and increase performance. Because of these considerations, all three methods (keyword detection, Shrestha and McKeown method, and Cong et al. method) were employed and compared.

The simple keyword detection covered the basic question indicating keywords 'wie', 'wat', 'waar', 'wanneer', 'waarom' and

Hey mensjes.als jullie Weleens eerder mijn profiel hebben aange-
 likt weten jullie misschien dat ik zo'n anderhalf jaar een jongen
 leuk begon te vindenwe hebben nou al 3x afgesproken, alleen
 nou ging t mis. De laatste x was ik bij hem en waren we aan
 het kletsen over vrienden en relaties en over mijn blauwtje bij
 hem precies een jaar terug op de ijsbaan.nu zullen jullie denken;
 ok leuk da verhaal maar wa motte we der mee? Nou... zo ong.
 Dit; Ik heb het gevoel dat we nu zo close als friends zijn geraakt
 dat het liefdesvuur is gedoofd. Niemand wil me geloven maar
 ik ben stiekem wel blij da'k der vanaf ben. Alleen als iedereen
 hem nou extra agaat plagen hij zijn vrienden zowat bij die op-
 merkingen zo'n stomp geeft dat ze zowat hun ribben kneuzen
 en dat we iedere x allebei knettergek worden van dat gekoppeld
 worden. Betekent dat dan dat ik serieus NA ANDERHALF JAAR
 GEWACHT TE HEBBEN MIJN GEVOELENS HEB VERLOREN
 PRECIES OP HET MOMENT DAT HIJ MIJ LEUK BEGINT TE
 VINDEN??? Ff laat ff weten wa jullie der van zeggen. Moet ik m'n
 vrienden geloven dat ik mijn gevoelens alleen wegdruk omdat ik
 er vanaf wil komen en dat niet lukt, of mij volgen omdat ik denk
 dat ik van de liefde genezen ben? AV bedanktmvg Fenkxps als
 jet nie snap snap ik da want t is lastig uit te leggen

Figure 17: Example of poor spelling and grammar, including contamination, spelling errors, grammatical errors and slang.

'hoe', which are 'who', 'what', 'where', 'when', 'why', and 'how' respectively. Additionally, question marks were detected. The presence of at least one keyword and a question mark will classify the post as containing a question.

The Shrestha and McKeown method was implemented by using the SpaCy model (Choi et al., 2015) trained on a Dutch corpus of news articles as this model is fast, available as a pretrained model with a Dutch dataset and performs reasonably well. A (linear) Support Vector Classifier is trained and used to classify the data. The SVC was trained on the subset of data which was annotated. As was done in the original study, only the first and last five POS tags will be used for each sentence. Sentences which are too short will be padded.

The Cong et al. method was implemented similarly to the Shrestha method, with a SpaCy model trained on the same Dutch corpus for part of speech tagging and the same Support Vector Classifier for classification. All words except the keywords listed in the simple keyword detection method were POS tagged. Unlike the Shrestha method, all words from every sentence are used.

The Shrestha et al. and Cong et al. methods were trained and tested in a ten-fold crossvalidation scheme, testing for both accuracy and MCC score. This was done once with all available features and once with the k-best features, with k=200 features using chi-square for feature selection.

The Cong et al. method with the 200 best features performed best out of all of these variations and is therefore the algorithm that is used in the reply resolution algorithm. See section 6.2 for more elaborate results.

The answers to the detected questions were determined by cosine similarity, using TF-IDF bag of word representations of the posts. Cosine similarity is one of the methods of answer detection referenced by Cong et al. (2008) and Shrestha and McKeown (2004) and, while the simplest of the proposed methods,

it is adequate and was chosen of methods which might have performed better because these methods were not feasible within the scope of the project.

The threshold for similarity was determined by balancing question-answer pair correctness with dependency correctness. If a threshold which is too high was used, average dependency correctness would suffer, as the final and last node in the tree is based purely on heuristics. However, if a threshold was chosen too low, performance might have been worse than the heuristics in the last node.

5.2.6 Proximity

According to Kim et al. (2010), corresponding posts tend to be temporally close to each other, this is specifically true for the posts by the original poster. They also claim that posts from non-initiators tend to not respond to one another. However, temporal proximity is not the only factor at play. As shown by Aumayr et al. (2011), similarity can be used to detect similar language usage, even if it is not a direct quote. To combine these heuristics, a back-off penalty is applied to the similarity scores of posts preceding the posts for which no antecedent has been found. This is implemented by subtracting $\log_{10}(k)$ where k is the reply distance from the similarity score. To account for the increased likelihood that a post is a response to the original post, this penalty is not applied to the original post. After this penalty, the post with the highest score is chosen as antecedent for the given document.

5.2.7 Evaluation

The algorithm was evaluated as whole, which means that the several components are not all evaluated independently. The only exception is the question answer pair component, which needed to be evaluated in order to choose the appropriate algorithm. The reply relations algorithm as a whole is evaluated by comparing the accuracy of the predicted labels with the average probability of the predictions. As the probability of correctly identifying the reply relation by chance differs per message because of the differing number of messages which came before it, the probability was calculated per classification and averaged over all messages.

5.3 Features

5.3.1 Empathy related features

The final selection of empathy related features in this study was made based on the eight constituents defined in section 2.1, the literature surrounding empathy in online contexts (section 2.2) and in youths (section 2.3) as well as the conducted interviews (section 5.1).

As suggested in the interviews, the eight constituents were seen as a scale ranging from low empathy (represented by merely cognitive insight) to high empathy (feeling for another because of the situation they are in). To represent the direction in which the feelings described by the several levels of empathy are going, a valence scale is added. This leads to a combined empathy-valence score which can be used to approximate empathetic responses.

Because of the online nature of the interactions on the forum, the only present cues for empathy is in the text of the messages. Many physical cues which would normally indicate empathy are absent, and with them the neurological mimicry as described in section 2.1.2. For this reason, the physical mimicry component is not included as a level in the empathy scale.

In the interviews, constituents four and five were consistently grouped because of their high amount of overlap. These constituents were also grouped as one level in the empathy scale in this study. Constituent seven and eight were also grouped but not consistently. The reasoning to keep these constituents separate is the origin of the feelings for another, which lies in the situation that the other is in for constituent seven but in the person itself for constituent eight. This reasoning was followed and constituent seven and eight were kept as distinct levels in the scale.

This yields the final features for presence of empathy:

- I understand that you feel this way
- I can imagine that you feel this way
- I would feel this way if I were in your shoes
- I feel this way too
- I feel this way because of the situation you are in
- I feel this way because of what you have been through

Along with valence levels to indicate which emotion should be filled-in in the empathy type description:

- happy
- cheerful
- apathetic
- touched
- hurt
- sad
- angry

5.3.2 Forum page features

For each thread, the thread name, tags, topic and best answer were saved. The thread name serves to group messages under one unified title and helps annotators determine context. The tags are user-generated and therefore have a large amount of variation. The variance in usage, in the amount and in the tags themselves is such that the tags were not used as a feature in the models. The topic serves to cluster the data but is also used as a feature in the models as messages in some topics are more likely to require an empathetic response than in others (for example 'emotional problems and feelings' vs 'sport and free time'). The best answer feature, though currently enabled, was disabled in the forum for an unknown period of time (Kindertelefoon, 2017). Because of this, an unknown amount of threads do not contain a 'best answer' although they might have if the feature was enabled. Because of this irregularity, the 'best answer' was not used as a feature.

For each message, the raw text is saved along with information about the user who wrote the message, likes, mentions, citations and the date. See figure 18 for the full list. The user ID as well as the username are saved because the website is not consistent in referring to a user by the username or ID alone. For example, user mentions in a message refer to a username while a citation of a previous message references only the user ID of the writer of the previous post. Mentions and citations of previous posts were collected as they are both highly indicative of the post to which the current message is a reply.

5.4 Models

Combined LSTM model. The combined LSTM model is based on the models implemented by Khanpour et al. (2017) and Saeidi et al. (2016), using the same LSTM layers and output generation as used in Khanpour et al. (2017) with the input from Saeidi et al.

Thread:

- Name
- Best answer
- Tags
- Topic

Message:

- Raw text
- Date
- Username
- User ID
- User title
- likes
- Mentions
- Citations

Figure 18: Features which were downloaded from the Kindertelefoon forum page.

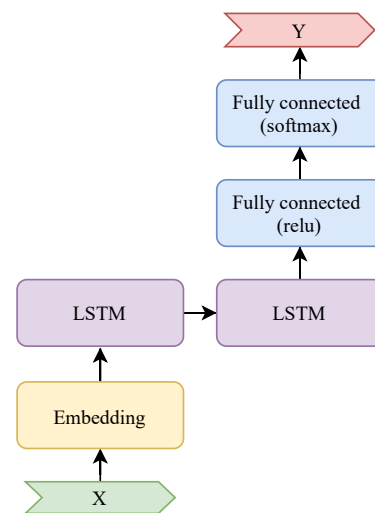


Figure 19: Layers of the LSTM model used in this study.

(2016). This combination was chosen because the output representation of Saeidi et al. (2016) does not match the classification output desired in this study and the convolutional layers from Khanpour et al. (2017) were considered superfluous, as they prevent the LSTM model from utilizing the raw embedding data from the documents.

The buildup of the LSTM model as it is implemented in this study can be seen in figure 19. The LSTM model used in this thesis uses an embedding layer as described in section 3.3, which projects the input sequence in the trained embedding space. This approach was chosen over a pretrained Word2Vec approach as the vocabulary of the Kindertelefoon data was not expected to match the Word2Vec training data. This embedding is processed by two LSTM layers. The output of the second LSTM layer is fed through a full connected RELU activated layer and finally through a softmax activated (see appendix E for more information about activation functions) fully connected layer to produce an output between 0 and 1 for each class, indicating the probability of the document belonging to that class.

BERT model in this study. The BERT model used in this study used the Dutch pretrained Bertje model (Vries et al., 2019) which is trained on Dutch fictional novels (4.4GB), Dutch news

Layer	Shape	Params
Embedding	(None, max. input length, 64)	1920000
LSTM	(None, max. input length, 100)	66000
LSTM	(None, 100)	80400
Fully connected	(None, 100)	10100
Fully connected	(None, 2)	202

Table 6: Shapes of the layers in the LSTM model as implemented in this study.

corpous TwNC (Ordelman et al., 2007) (2.4GB), the SoNaR-500 corpus (Oostdijk et al., 2013) (2.2GB), various internet news (1.6GB) and Wikipedia (1.5GB). The pretrained BERT model was used in conjunction with a fully connected layer which, though simple, has proven to be capable of natural language understanding in this model (Li et al., 2019).

On top of the approximately 40 epochs of pretraining of the Bertje model data, the same pretraining tasks were applied with the full Kindertelefoon dataset. The tasks in the additional pretraining followed the Bertje documentation exactly, so the model is training using the exact same tasks, except on the Kindertelefoon data. This additional pretraining was performed to enable the model to get a better understanding of the domain language of the Kindertelefoon data.

As the Bertje developers have used slight alterations to the NSP and MLM tasks, the same alterations were used in the training tasks for the additional pretraining as well. Whereas the original BERT developers used arbitrary length text chunks as sentences for next sentence prediction, the sentences used in the Bertje model were actual sentences. In the original BERT model, sentence B was a random text chunk from the same document if it was not the next sentence to sentence A. This means that sentence B may be very different from sentence A, as it may be taken from an entirely unrelated portion of the text. The Bertje authors considered this task too easy and have defined sentence B to be either the next sentence or the previous sentence, which is known as the Sentence Order Prediction (SOP) task.

The BERT authors have updated the MLM task since the first publication of BERT, which implemented the masking of entire words instead of wordpiece tokens. This makes the task considerably more difficult, as wordpiece tokens could be guessed with relative ease based on the surrounding wordpieces, as the same words always produce the same wordpieces. The Bertje developers have included the whole word masking in the MLM task, and so it has been included in the additional pretraining as well.

The model was fine-tuned using the combined annotated data, during which the classification layer was trained. Devlin et al. (2019) advised two to four epochs for finetuning. The model was tested in all three epochs in this range, as both too few and too many iterations might yield poor performance. In the case of too few iterations, there might be room for improvement if given more opportunity to learn and in the case of too many iterations, the model might overfit the data.

During finetuning, the pretrained layers were updated along with the classification layer. This was done to enable the encoding to be optimized for the classification task. Another set of models was run with the transformer layers frozen during the classification task, which enabled the effect of the pretraining to be identified better.

The maximum input sequence length for BERT-base on which Bertje is based is 512 tokens, of which the [CLS], and the two [SEP] tokens take up three. For the pretraining MLM task, the sentence pairs were truncated such that their combined length did not exceed 509 tokens. The sentences were truncated both left-sided and right-sided with a 0.5 probability of either side. During finetuning, documents were truncated to a length of 507 tokens, to leave room for the [CLS] token and [SEP] tokens, as well as the additional usertitle and topic features which were placed in sentence B.

For the BERT model used in this study, version 3.0.2 of the transformers library for python was used. Version 4.x of the transformers library has breaking changes for the script. Because of this, cuda 10.1 and tensorflow 2.3.x should be used.

Model comparison metrics Because the MCC score is less sensitive to imbalanced data, it is used as main evaluation metric for the models used in this thesis. To facilitate comparison with other metrics, the MCC score is normalized by mapping the values linearly between 0 and 1, which results in a score of 0 for perfectly wrong predictions, 0.5 for chance level predictions and a score of 1 for perfect predictions. To facilitate comparison with existing models which often use *F1* scores, the *F1* scores are included in the results.

Call for empathy Using the models to detect empathetic responses in combination with the reply relations determined earlier, the messages which call for an empathetic responses were classified. The accuracy of the combination of the reply relations algorithm with both the BERT model and the LSTM model was determined using the annotated ‘call for empathy’ label. The best performing version of the model will be used for this determining the empathy labels. Each annotated message was first classified into empathetic or non-empathetic classes. For every message, the antecedent was determined through the reply relation algorithm. The number of replies which is empathetic was counted for every message. Binary labels were made with a number of thresholds for this count value, up to the maximum count of any message. Normalized MCC scores are then calculated for every threshold value for both models.

Aside from the approach using the empathy labels and reply relations, call for empathy was modelled directly using the BERT and LSTM model. As was for the classification of empathetic responses, the models trained to classify call for empathy were trained on non-oversampled data and oversampled data. For the BERT model, again all pretrain conditions including no pretraining were used as starter model.

5.5 Data collection¹⁰

To train the models used to classify posts which call for an empathetic response, data and metadata is needed. The main data takes the form of the messages which are posted on the Kindertelefoon forum. To perform supervised training, information about these messages is needed. This information is gathered through annotation of the message data. This will result in a dataset with annotated labels for each text.

The data for this project was collected by systematically crawling every thread on the forum and saving the features listed in

¹⁰Data collection for this thesis was planned and executed as part of a separate project. This was done to be able to allocate more resources in collecting data which would enable a better study as main thesis. This division of a part of the project means that some of the documentation provided for the data collection in the Advanced Research Project report will be present here as part of this thesis as it is part of the same project.

figure 18 as there was no direct database or API access. A total of 30494 threads were downloaded, containing 221707 messages by at least 11545 users. As an unknown amount of users will have turned 18 and had their account removed, the number of users is higher.

The source code for the scraper, dependency builder and empathy classification algorithm can be found on <https://github.com/lucas-su/empathy-tagger>.

5.5.1 Scraper architecture

The first iteration of the scraper was made such that it visits the user overview page for each post on a given thread, after which it continues scraping the first page of each thread that the visited user has posted on. This method sprawls out quickly but might miss threads which are only responded to by users who have not responded to any other thread indexed by the scraper. In addition to this architecture possibly leading to missing data, it took increasingly long to check whether a thread had already been downloaded. Hence the architecture of the scraper has been changed to index all threads through the ‘all threads’ overview pages for every topic, saving every link to a thread and then visiting them individually.

5.5.2 Error handling

A number of different unexpected errors came to light while scraping. Threads of which the topic has changed are not correctly scraped beyond the first page, as only the first page correctly forwards the session to the new URL with the new topic. There have been 151 threads which have changed topics, all of which have been checked for completeness manually.

Four users whose names have been changed were not indexed because an internal server error (500) was returned when visiting the user overview page. This was not the case for all users who have had their name changed, as there were at least 546 requests to change a username (Kindertelefoon, 2015).

A total of four threads returned the 503 (unavailable) error code. These threads are thought to be deleted by the Kindertelefoon. This does not normally happen but may occur if the content of these threads is offensive or harmful¹¹. Two other threads return a 404 error, they cannot be found even though they are listed in users’ activity overviews. These six threads were not indexed by the scraper.

5.6 Annotation

To be able to train the language models and assess the performance of the models and the dependency building algorithm, a subset of the data was annotated. A total of 1500 threads were selected by weighted random selection from the full dataset. The weights for this selection were based on the relative frequency with which each topic was featured in the full dataset. This provides a similar distribution of topics within the annotation dataset with regards to the full dataset. Within each topic, the threads were randomly chosen. Of the 1500 chosen threads, 18 contained over 100 messages. These very long threads are unlikely to be fully annotated by annotators before they are no longer willing to continue and are hence omitted.

Figure 20 shows the proportions of each topic in both the full dataset and the annotation dataset. Each topic is represented by at least one thread in the annotation dataset and ratios are

close, the largest absolute difference between full dataset and the annotation dataset is 0.025 percent point.

5.6.1 Annotation labels

To be able to classify posts which call for empathy through posts which express empathy, a number of labels need to be defined. The ‘is empathy’ and ‘call for empathy’ labels are the main two variables, denoting the presence of an empathetic response and the call for such a response respectively. To be able to distinguish between the variance in amount and valence of empathetic responses, empathy amount and empathy valence options are constructed based on the empathy literature study (see section 5.3.1).

To resolve which post is a response to which previous post using the post relation algorithm (see section 5.2), the label ‘response to’ was used, as well as ‘is question’ and ‘answers question’.

In the interviews with the psychologists (see section 5.1), the lack of a call to action label came to light. Many responses on the Kindertelefoon forum give advice on how to deal with certain situations in a practical manner. To incorporate the call to action component, a ‘call to action’ label has been added. Figure 21 summarizes the labels which need to be annotated.

The process of collecting data from the Kindertelefoon forum as well as that of the annotation of the data was reviewed by the ethics committee of the faculty of Electrical Engineering, Mathematics and Computer Science of the University of Twente. The scraping of data from the forum was done in consultation with and with approval of the Kindertelefoon. The privacy statement of the Kindertelefoon states that all posted messages on the forum constitute public data and may be seen or used by anyone. Additionally, it explicitly states that data posted to the forum may be used in scientific research.

5.6.2 Annotation page layout

The first iteration of the annotation program was a terminal based application (see appendix D for a screenshot). To enable others to more easily annotate data, the second iteration of the annotator program was made as a website which can be seen in appendix D. Functionally, the program is similar except for a small number of additional prompts.

The landing page constituted the information brochure for the study and stated that annotators may stop participating at any point in time, that questions about the study or content of the messages may be addressed to the study coordinator and that questions which cannot be sent to the study coordinator may be sent to the supervisor. Annotators were instructed on how to annotate the data, what the goal of the study was and how they were to contact the study coordinator in case of inappropriate content, questions or discontinuation of participation.

The page provided the annotators with new messages to annotate as long as they kept submitting annotations. When, at some point, the annotators chose to stop annotating, they would simply close the website. The remaining messages of that thread were not assigned to another person, as it would be confusing to start annotating in the middle of a thread. The entire thread was also not assigned to another person, as this might lead to more than three annotations per post if the initial user were to continue. Because of this, some posts have only two annotations.

The annotator was asked to fill in a username, which was used to keep track of which annotator has covered which threads. As

¹¹Titles reference for example self harm explicitly which is an indication for this.

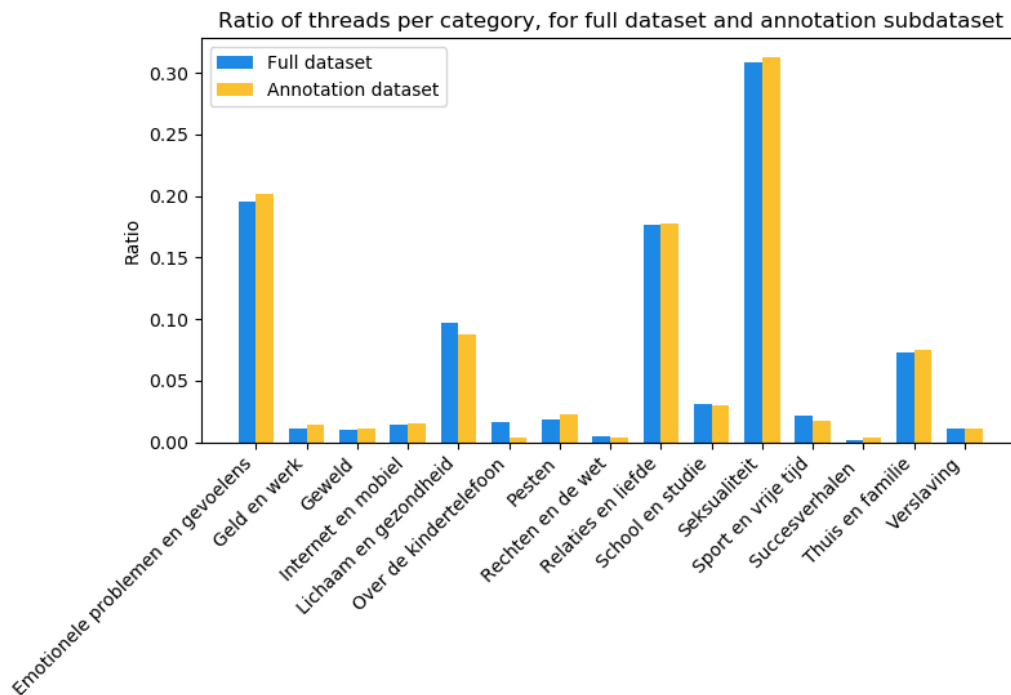


Figure 20: Comparison of ratios of topics within the full dataset and annotation dataset.

- This post expresses empathy
 - Empathy amount:*
 - I understand that you feel this way
 - I can imagine that you feel this way
 - I would feel this way if I were in your shoes
 - I feel this way too
 - I feel this way because of the situation you are in
 - I feel this way because of what you have been through
 - Empathy valence:*
 - happy
 - cheerful
 - apathetic
 - touched
 - hurt
 - sad
 - angry
- This post calls for empathy
- This post asks a question
- This post answers a question
- This post proposes an action
- This post is a response to post x

Figure 21: Labels which need to be annotated. Items marked - are mutually exclusive within their category, items marked • are not. The question labeled ■ is an integer input, all other are binary inputs.

each thread was annotated in threefold, a username was necessary to prevent one user from annotating the same message multiple times. The username was stored in a cookie on the user's device, and this username was automatically used if a user

stopped and decided to continue annotating at a later time. Users were required to explicitly agree with the placement of the cookie before participating as required by the GDPR.

Threads were presented one post at the time, but previously annotated posts within a thread were always visible. Users were presented with the statements listed in figure 21. The statements marked • were binary non-exclusive inputs (checkboxes), and were to be checked if the statement in question was appropriate for the post which the user was annotating at that time. If a user indicated that a post was an expression of empathy, they were asked what kind of empathetic expression was used there. Users indicated the amount of empathy in the message and the valence of the response with the ultimate aim to mimic the post they are reviewing as closely as possible with the given options. The options for empathy amount and empathy valence are marked with a dash (-) in figure 21 and are mutually exclusive in their category (radio buttons).

After choosing two options, an example message appeared using the two selected options. If for example option 3 was chosen for the empathy type and option 7 was chosen for valence, the example sentence would be: *I would feel angry if I were in your shoes*. This sentence could be compared to the message which was reviewed at that time. If the general gist of the message matched the example sentence, the user could continue.

If the message which was being annotated at that time was the third message or a later message, users were asked to indicate to which previous message the message they were annotating at the time was a response. This question was only asked from the third message on, as the second message could only be a response to the first and the first message was not a response to any previous message. The antecedent to the post which was annotated at the that time could be indicated through either a number input field at the bottom of the page in which the number of the antecedent

post should be given or by clicking on the antecedent post. If an antecedent is indicated through either way, that post will be highlighted in green and the post number will appear in the number input field at the bottom of the page.

Messages by users who were deleted from the forum were saved with username ‘anonymous’ and user id 0. To make it clear to annotators that this was not in fact one user with the name ‘anonymous’, this name was displayed on the annotation page as ‘a deleted user’.

Every thread was annotated by three annotators. The threads were presented in order of appearance in the annotation dataset, which ensured that the threads which were annotated were all annotated by three annotators. If not enough annotators could be found, at least those threads which were annotated were annotated by three annotators.

5.6.3 Pilot test

Before reaching out to annotators, a pilot test was conducted, in which a number of threads were annotated and thoroughly checked for correct processing by the server. During the pilot test, multiple annotators used the website simultaneously to ensure that the annotations are attributed to the correct annotator. Apart from functional testing, there was much feedback on the user interface.

To differentiate the post which the user is annotating from the previous posts, this post needed to be highlighted better. This was initially done by marking it green, but this was found to be annoying, especially on mobile devices and for longer texts. Instead, a green arrow is used to indicate the current post and reply icons are used to indicate previous posts in addition to the textual descriptions. The green highlight used to indicate which previous post was selected as antecedent was not conceived as interfering, as this message was not read with as much attention as the post which was annotated at that time.

Initially, there was a default selection for the antecedent, which was the previous post. This was correct often, but naturally not always. When this was not correct, it led to errors during testing as users forgot to correct the default value. In response to this, default was removed. The antecedent had to be indicated manually for every post.

To make the back button more easily accessible and to make it more obvious that users could go back to previous posts, a back button was placed under the save button in addition to the back button at the top of page and the browser back button.

5.6.4 Platforms

Participants were asked to voluntarily participate in the annotation through the social media platforms LinkedIn, Reddit and Facebook, personal direct messaging channels, the human research participant platform SONA and through paid platforms. The SONA participant pool is internal to the University of Twente and serves to provide studies from the social sciences with participants by requiring students to participate as part of their curriculum.

A link to the survey was posted on [reddit.com/r/SampleSize](https://www.reddit.com/r/SampleSize) and [reddit.com/r/takemysurvey](https://www.reddit.com/r/takemysurvey), which are communities in which users participate in academic studies voluntarily out of interest for academic research.

Two of the available paid crowdsourcing services are large enough to be able to provide Dutch speaking participants, which

is a requirement for this study. These two platforms are Amazon’s Mechanical Turk (MTurk) and Prolific Academic, which is a participant pool set up by a number of British universities. In a comparison between the two paid participant platforms, Peer et al. (2017) have found the participants in Prolific Academic’s pool to be more naive, more diverse in their ethnicity, more attentive and more honest (Peer et al., 2017). However, due to the larger overall pool of participants, MTurk offers faster response times and may, despite a lower diversity in ethnicity and locales, offer more participants who speak the Dutch language because of the same reason. Due to financial and logistical considerations, the annotation questionnaire was only published on Amazon’s Mechanical Turk platform.

Mechanical Turk participants who did not use their MTurk user-id and did not enter the correct password shown on the website were rejected, as their work could not be verified. Workers who filled-in the correct password but did not use their MTurk user-id were asked to present their username as used on the website to verify their work.

5.7 Data processing

Because the number of spelling errors in the data were a concern for a number of steps in the study, the number of misspelled words in the full dataset were counted. A dictionary compiled by the Opentaal foundation was used to look up words in the texts. This check was limited to spelling errors, grammatical errors were not included.

After data collection, the threefold annotations were combined into a single ground truth to be used to train and evaluate the models. For the binary labels, the most frequent label was chosen. For the posts with two annotators, the statement was assumed to be true in a true/false tie. The empathy amount and empathy valence labels were selected as follows: if one label occurs at least two times, that label was chosen, else, the rounded average of the scores was chosen. For example, if for empathy amount option 1, 2, and 6 were chosen, the ground truth value would be 3. If 1, 1, and 6 were chosen however, the value would be 1 as there was a majority.

Users who were quoted in messages were added to the ‘mentions’ list, as a quote is considered a mention as well.

As the proportion of texts labeled as empathetic is very small (just under 5%), artificial text generation was used to oversample the class labeled ‘empathetic’. Using Dutch word embeddings produced in (Tulkens et al., 2016), a semantic average of the embedding vectors of each combination of two texts labeled as containing an empathetic response were created. The embeddings used are based on tokens taken from web data which was scraped from .nl and .be domain websites. The web-based (Corpus from Web, CoW) embeddings were chosen as these are most likely to be representative of the Kindertelefoon data and are more likely to match the vocabulary used on the forum than a corpus based on tradition media or wikipedia. Additionally, Tulkens et al. (2016) find that the COW embeddings outperform even combined datasets from other sources such as the Roularta, SoNaR and Wikipedia corpora. The text generation yielded 6388 new texts which were added to the dataset to bring the total proportion of empathy labeled texts to exactly half. An example of such a synthetically generated text can be seen in figure 22.

· het spijt me enorm wat ik goed hoop dat dat zou het zo sterk
 · een jongen van snapchat plek waar niet betrap je echte moet maakt het
 · je ziek blijf allemaal heel thuis is en natuurlijk mogelijk te geen anderen te de mensen van je jullie gaan roddelen over je ik dat persoonlijk ook een heel aangezien je snap dat je dit naar wil was je kan elkaar en aanraken etc. verklaring te wel heel gedurfd er echt wil gewoon ik ongezonde

Figure 22: Examples of a synthetic text as produced by the synthetic minority oversampling technique.

5.7.1 Annotator agreement

As the three annotators for each text are not necessarily the same three annotators, regular interrater reliability metrics cannot be used. Instead, the agreement between the annotator and each co-annotator was used as a metric of annotator quality. For each annotator, the number of times another annotator agreed with the label was counted. This proportion of agreement was calculated for each binary label. The metric was not applied to the empathy type and empathy valence, as these measures are dependent on the “is empathy” and are therefore not suited for this comparison. The agreement scores are calculated in multiple iterations, where the mean agreement of the last two iterations forms a weight for calculating the agreement for the current iteration. Because of this weight, the annotators with a high agreement score, in other words the annotators which are likely to be good annotators, will weigh in more heavily in determining the agreement score for the next iteration. Agreement with annotators which are likely to be good increases the agreement score because the likelihood that the agreeing annotator is also good is high when agreeing often with good annotators. The formula for the weight W for iteration i is expressed as follows:

$$W_i = \text{mean}(2a_{i-1}, 2a_{i-2}) \quad (12)$$

For the first iterations, $i - 1$ and $i - 2$ are 1 for each annotator.

Each iteration, the annotators which have a agreement score lower than one standard deviation from the mean agreement are manually inspected. The annotators which are assessed to be poor annotators by this manual inspection are not included in the next iteration, until no annotator with an agreement score below one standard deviation from the mean fails the manual inspection.

6 Results

The results giving insight into the five main research questions are presented in this section. The annotator reliability assessment is discussed in section 6.1. This allows the first research question - *Which annotators are reliable and which are not?* - to be answered.

The performance of the components of the reply relation algorithm are presented in section 6.2, with which the second research question - *How well can the different components from the reply relations algorithm assess the reply relations?* - can be answered.

The pretraining metrics for the BERT models are presented in section 6.3, which do not on their own answer a research question but are relevant for research question three.

Data pertaining to the third research question - *How well do the LSTM and BERT model classify empathy?* - is presented in

section 6.4. Both oversampled and non-oversampled datasets are covered here. For the set of BERT models, models based on different pretraining epochs are compared to be able to answer the subquestion concerning the impact of the pretraining epochs on the empathy classification performance, both in a version with transformer layers frozen in the empathy detection training and a version with trainable transformer layers.

The performance of the combined reply relation algorithm and the empathy classification models to classify a call for empathy is presented in section 6.5.1. This enables the forth question - *How well does the combination of empathy prediction and reply relation algorithm work?* - to be answered.

Performance metrics of the BERT and LSTM models tested on call for empathy directly in order to answer the final and exploratory research question - *How well do the LSTM and BERT model classify call for empathy directly?* - are presented in sections 6.5. As was the case for the empathy classification with these models, both oversampled and non-oversampled datasets are covered. For the set of BERT models, models based on different pretraining epochs are compared to be able to answer the subquestion concerning the impact of the pretraining epochs on the call for empathy classification performance, both in a version with transformer layers frozen in the call for empathy detection training and a version with trainable transformer layers.

6.1 Annotators and agreement

Figures 23 and 24 show graphs of interannotator agreement scores for the first and last iteration of annotator removal. The graphs of the intermediate iterations are available in appendix C. The first iteration of the annotator agreement assessment yielded a mean agreement between annotators of 0.55. There were 35 annotators who scored below one standard deviation from the mean of which 6 annotators scored below two standard deviations from the mean. After manual assessment of the annotations, all of the 35 annotators were removed from the dataset. This data is shown in figure 23.

The 134 annotators left in the dataset after iteration one were assessed in the second iteration, in which 22 annotators scored at least one standard deviation from the mean, one of which scored below two standard deviations from the mean. The annotations of the 22 annotators who had below average agreement were manually assessed and the annotations of 20 of the annotators were removed from the dataset. This data can be seen in the lower half of figure 49 in appendix C.

The remaining 114 annotators were assessed in the third iteration of assessment. This yielded a further 15 annotators which scored below average and manually inspected, which is shown in the upper half of figure 50 in appendix C. After removal of 14 of these 15 annotators, the final agreement scores were calculated which are shown in figure 24. The annotator which was not removed from the 15 annotators which scored poorly was one of the two annotators which were kept in the first iteration despite a low score, hence the total number of annotators which were kept in the dataset despite a score below one standard deviation from the mean is two.

It is notable that due to the weights applied between iterations, the agreement scores can supersede 1 after the first iteration. Because of the removal of the poorly performing annotators with a low agreement score, the mean also moves steadily up over iterations.

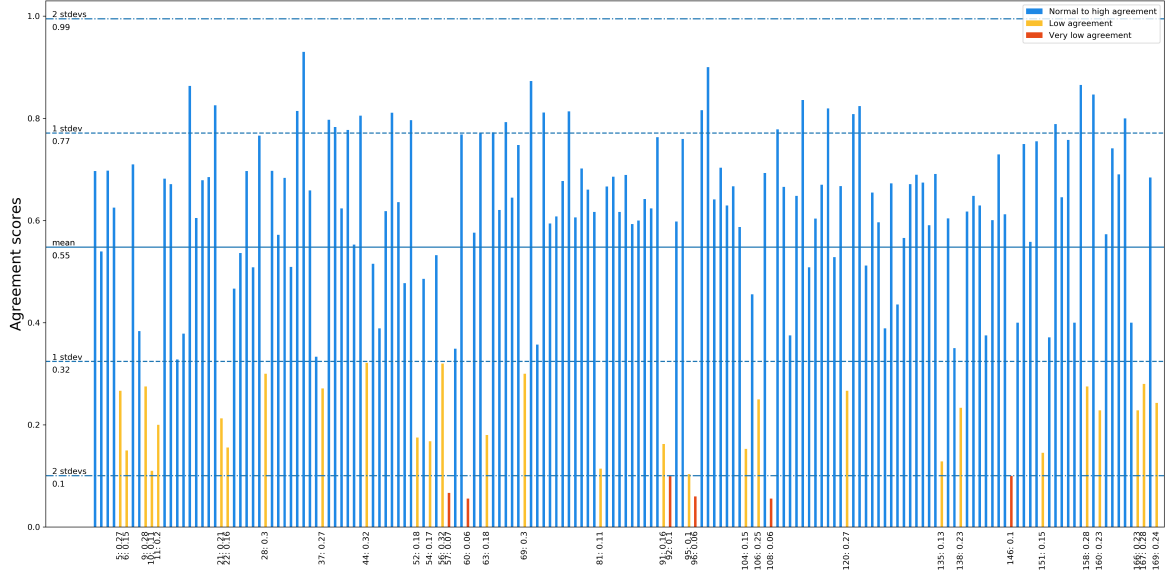


Figure 23: Agreement scores for iteration 1

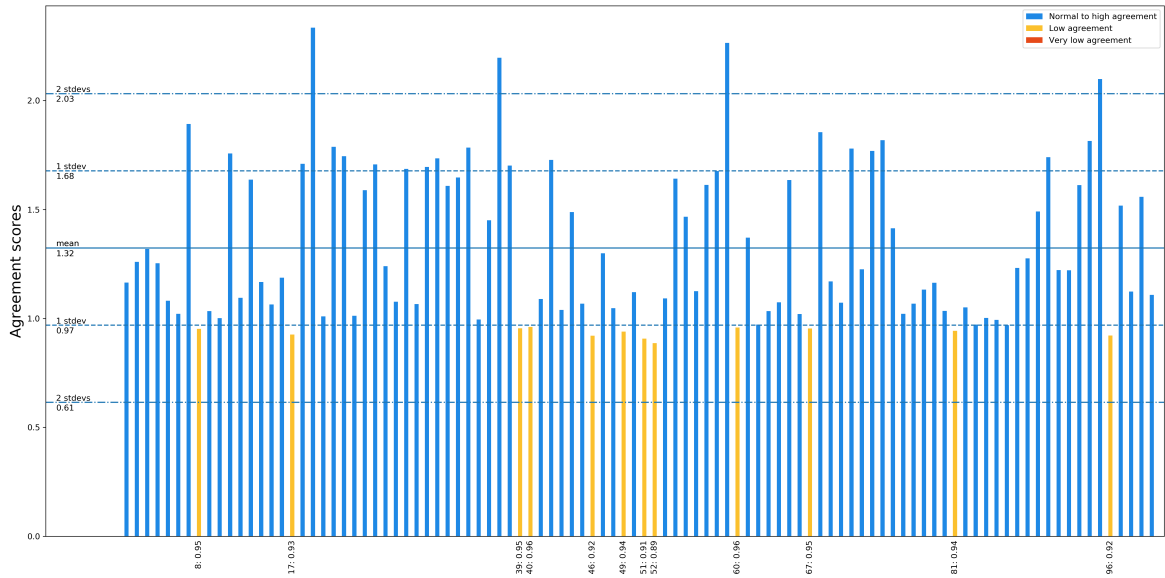


Figure 24: Agreement scores for iteration 4

Figure 25 shows the distribution of annotated messages among annotators before and after annotator removal. In the starting dataset, a total of fifty-two annotators covered less than 10 messages each, which is shown in the first spike. The spike starting off at 180 messages comes from Sona and Mechanical Turk participants who finished one session of annotation. The (small) peaks at 370 and 720 are from the same participants finishing their second and fourth sessions. The comparison between the frequencies before and after removal show that many of the removed annotators annotated less than 10 messages.

6.2 Dependency builder

The algorithm as a whole had an accuracy of 0.46 for all predictions. The average probability of classifying a reply relation correctly is also 0.46.

The proximity based similarity component contributed by far the most labels at 88% of the reply relations of the total algorithm, with the rest of the components contributing 0 to 5%. The accuracy of the different components varies between 0.32 and 0.48, and the prior probabilities of correctly labeling the messages labeled by each component varies between 0.21 and 0.81. The Z scores indicate that there was no significant improvement over chance level for any of the components or for the algorithm as a whole. The proportions of contributions, average probabilities

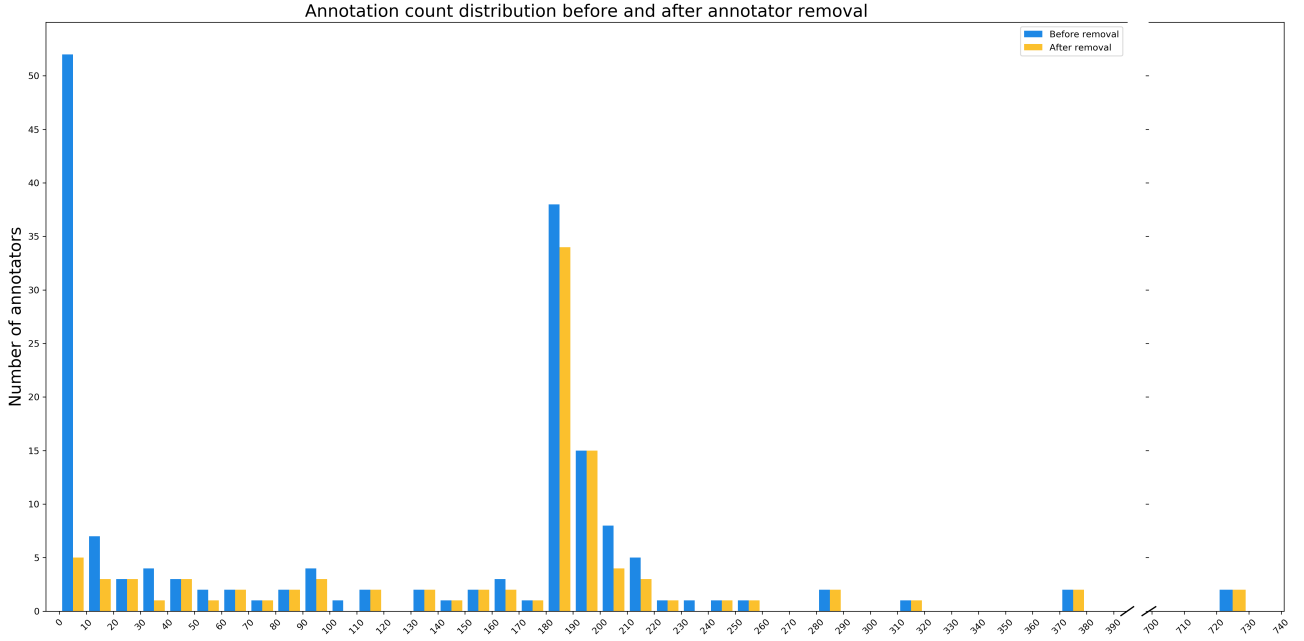


Figure 25: Annotation count distribution before and after annotators were removed. Note the broken x-axis around the 400 bin.

	% of labels	Avg. prob.	acc	Z score	P value
Is list	0.04	0.21	0.39	0.62	0.54
Actual mention	0.02	0.35	0.41	0.22	0.83
Similarity	0.05	0.226	0.32	0.39	0.70
Implied mention	0	0	0	N/A	N/A
Question-answer pairs	0.01	0.808	0.47	-1.01	0.31
Proximity based similarity	0.88	0.481	0.48	-0.003	0.998
All	1	0.46	0.46	0.005	0.996

Table 7: Proportion of total messages labeled by component, average probabilities, accuracy, Z-scores and associated p-values of dependency builder components.

	Accuracy		MCC	
	All features	kbest features	All features	kbest features
Simple	0.56	N/A	0.50	N/A
Shrestha et al.	0.66	0.67	0.49	0.50
Cong et al.	0.62	0.70	0.49	0.56

Table 8: Metrics of the three question detection algorithms tested. Values for the Shrestha et al. and Cong et al. methods are averages of the ten-fold crossvalidation results.

and performance metrics of each component can be found in table 7.

The question detection algorithm in the question-answer pair component of the reply resolution algorithm is the most complex element in the chain, which is why it is the only component which is evaluated separately from the other components. The

annotations included labels to indicate whether a post is a question or answer for this purpose. Table 8 shows the accuracy and MCC score for all features and the 200 best features for all three question answer detection algorithms which were tested. The simple keyword search was applied to all data, the Shrestha et al. and Cong et al. values are averages of the ten iterations in a ten-fold crossvalidation evaluation. In general, limiting features to the 200 best features yielded better results. Both in terms of accuracy and MCC score, the Cong et al. method performs best, with an accuracy of 0.70 and an MCC score of 0.56. For this reason, this is the method used in the final dependency builder architecture.

6.3 BERT pretraining

BERT was pretrained for ten epochs on all downloaded data from the Kindertelefoon forum. The same unsupervised pretraining tasks as the Bertje model it is based on were used, which are the sentence order prediction (SOP) task and the masked language model (MLM) task.

The upper half of figure 26 shows the accuracy on the final predictions on the ten epochs of pretraining as well as the baseline Bertje model. Both accuracy on the training set and validation set increase sharply after the first training epoch compared to the baseline. After the first epoch, the accuracy remains constant.

The lower half of figure 26 shows the training and validation losses over the ten epochs on the raw predictions of the model during pretraining. Epoch 0 represents the model before the additional pretraining tasks which were run on the Kindertelefoon data. Because the training loss from the Bertje model is not documented, only validation loss can be presented for epoch 0. The validation loss decreases sharply from the baseline Bertje model to the first iteration of training on the Kindertelefoon data, and then rises steadily every subsequent training epoch. The training loss decreases steadily every epoch starting from

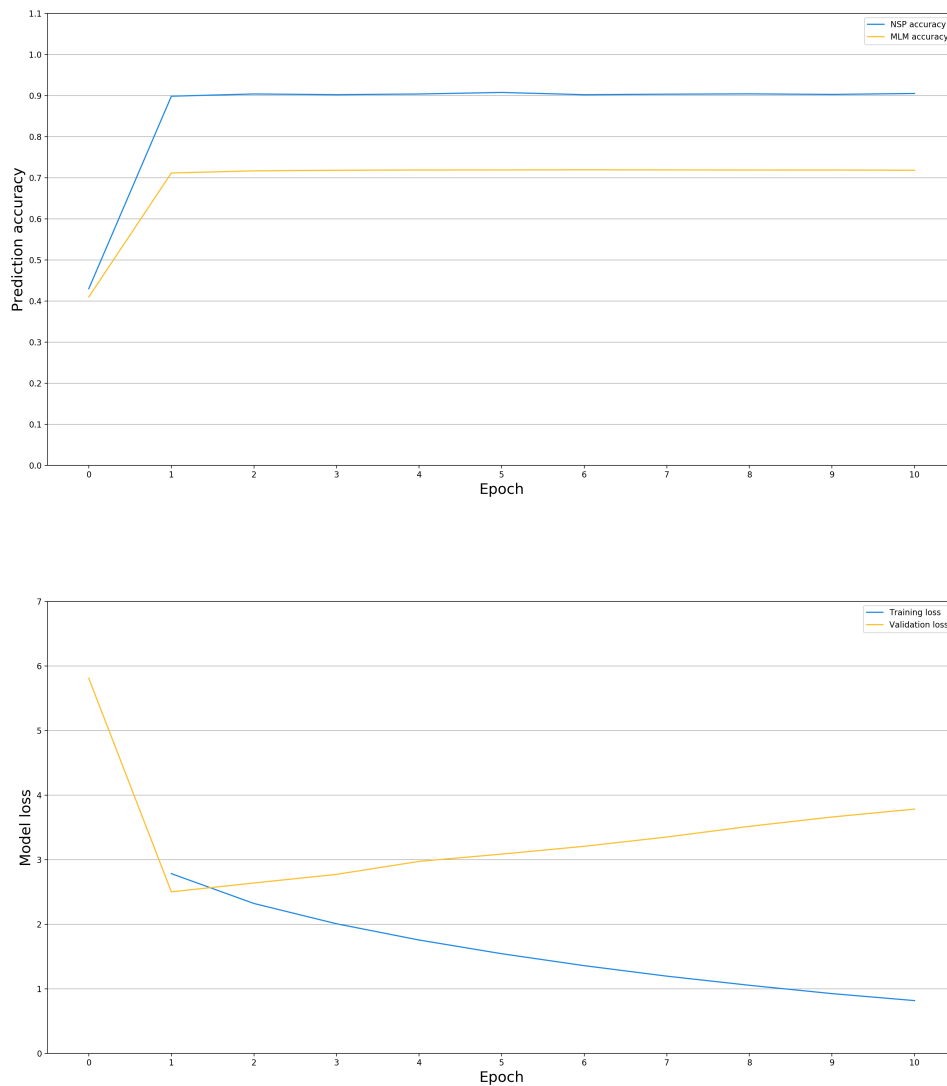


Figure 26: Prediction accuracy (upper) and model losses (lower) for pretraining tasks on the BERT model. Epoch 0 is the Bertje model before additional pretraining.

the first training epoch. After the first epoch, the training loss is slightly higher than the validation loss but from the second epoch on the validation loss is higher.

6.4 Empathy detection

Both the BERT and LSTM architectures were trained on non-oversampled and oversampled data, which are visualized in the graphs in this section as solid and dashed lines respectively. The models trained on non-oversampled data are referred to as NOS condition models for convenience, and the models trained on oversampled data as models in the OS condition. All models were trained in a fivefold crossvalidation scheme, the graphs shown are averages of the five folds. The data represented does not show performance of an individual model but rather the estimated performance of the architecture. The BERT models were trained with trainable transformer layers and with transformer layers

frozen after pretraining. In both cases, the classification head was trained on the classification task.

6.4.1 BERT

Because the BERT model can handle a maximum of 512 tokens on its input, messages longer than 507 tokens were truncated before use in the BERT model to leave room for the [CLS] and [SEP] tokens as well as the additional input features. Truncation was performed on 262 messages which constitutes 2.6 percent of the messages. For comparison, the percentage of messages which is too long is similar for the full dataset, at 5040 of the in total 221707 messages, constituting 2.2 percent.

Models with trainable transformer layers. The performance metrics of the NOS models are all fairly similar, with a nearly constant accuracy around 0.98 and an MCC score which is more noisy and hovers around 0.90 with a maximum of 0.93 and a minimum of 0.86. While all pretrain conditions hover around

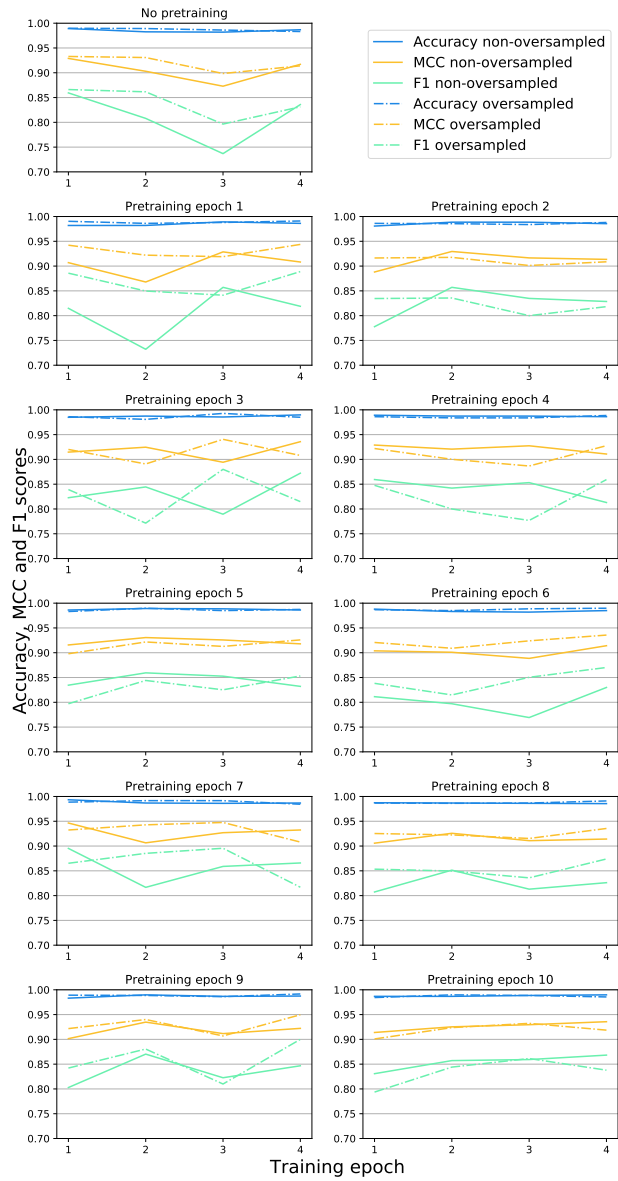


Figure 27: Validation accuracy, MCC, and F1 scores for the BERT model for empathy classification over training epochs, for every epoch of model pretraining. Transformer layers were trained with the classification head on the empathy detection task.

this value, there is no clear trend within the four training epochs of the shape of this variation. The F1 scores follow roughly the same pattern as the MCC scores but lower. Notably, the model based on one epoch of pretraining does not perform better than the model trained with no pretraining. Figure 27 shows validation accuracy, normalized MCC values and F1 scores for the four training epochs for all ten pretrain models and for a model

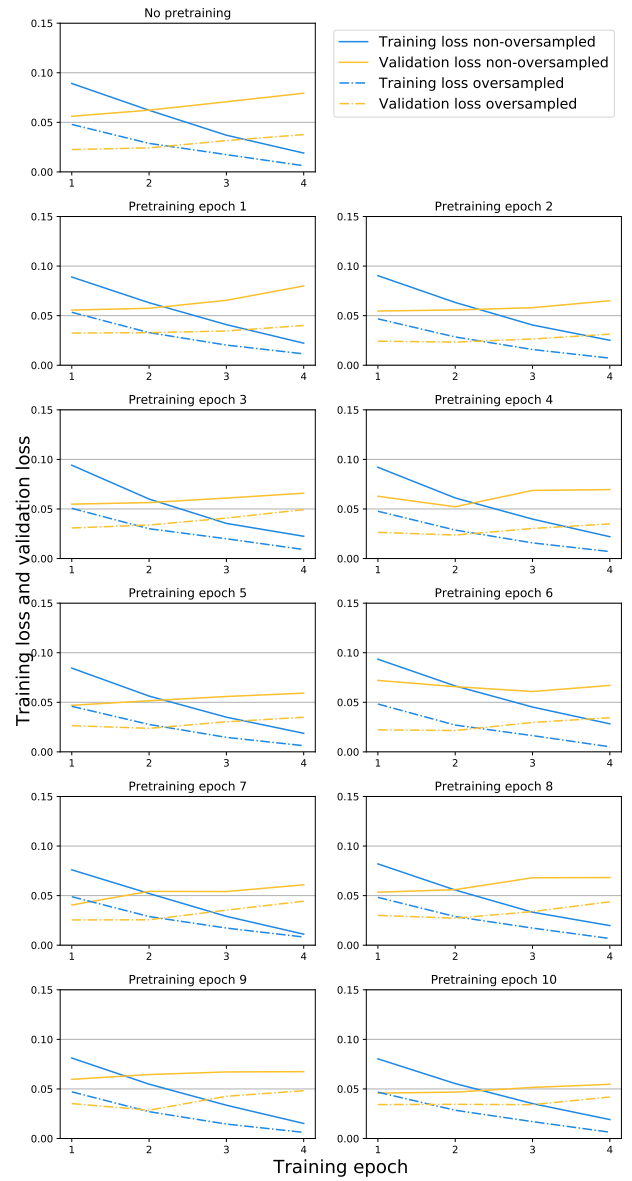


Figure 28: Model losses for the BERT model for empathy classification over training epochs, for every epoch of model pretraining. Transformer layers were trained with the classification head on the empathy detection task.

without pretraining. The graphs show the models trained on non-oversampled data (NOS models) and oversampled data (OS models).

The NOS models all show a similar pattern in losses. Over all four epochs, the training loss decreases steadily and the validation loss remains constant or increases slightly. The validation loss for the models based on no pretraining and on one epoch of pretraining increased somewhat sharper. At the first training

epoch, the training loss is higher than the validation loss. After epoch 2, the validation loss increases and the training loss decreases, such that the validation loss is higher than the training loss after epoch 2. Figure 28 depicts the train and validation loss for the four training epochs for all ten pretrain models and for a model without pretraining. Both the NOS models and the OS models are shown.

The performance metrics of the OS models are, in general, similar to the metrics of the NOS model. The accuracy is overall nearly constant and very high, similarly to the accuracy of the NOS models. The MCC scores and F1 scores of the OS models follow the same general trend of the MCC and F1 scores of the NOS models within most of the pretrain conditions, but are somewhat higher for the models based on no pretraining and on one epoch of pretraining. In the other pretraining conditions, the performance metrics of the OS models did not differ from the metrics of the NOS models. As was the case for the NOS models, the model based on no pretraining does not perform better than the model based on one epoch of pretraining.

The training and validation loss values for all the OS models follows the same pattern as the loss for the NOS models, but is overall lower, on average half of the loss values for the NOS models.

Models with transformer layers frozen. The models were also run without updating the transformer layers during the training task, relying fully on the pretraining for the natural language understanding training of the transformer layers. Figure 29 shows validation accuracy, normalized MCC values and F1 scores for the four training epochs for all ten pretrain models and for a model without pretraining where the transformer layers have been frozen during empathy classification training. The graphs show the NOS models and OS models, and are averages of the five folds of the fivefold crossvalidation scheme. Figure 30 shows the corresponding loss values.

The accuracy scores of the NOS models are nearly constant and hover around 0.95. The MCC scores for the NOS models are almost exactly constant with a value of 0.5, as are the F1 scores with a value of 0.

The loss values for the NOS models all follow a similar pattern. The training loss decreases between epoch 1 and 2, and remains fairly constant after epoch 2, around a value of 0.2. Only in the model based on the base model without pretraining is the loss lower, at a value of 0.18. The validation losses for the NOS models remain fairly constant at around 0.2, with again only the model based on the base model without pretraining having a lower loss, at around 0.18. The training loss and validation loss are about equal after epoch 2.

The accuracy scores for the OS models are all nearly constant with a value of around 0.9, except for the model based on no pre-training, where the accuracy was slightly higher with a constant value of 0.94. The MCC scores are all constant with a value of 0.5, the F1 scores are mostly constant with a value of 0.05.

The loss values of the OS models are slightly higher than the loss values of the NOS models. The training loss for all OS models decreases between epoch 1 and 2, and decreases at a slower rate after epoch 2. The validation loss starts out lower than the training loss and also steadily decreases, steeper between epoch 1 and 2 than after epoch 2. The validation loss is still decreasing, though slowly, and still lower than the training loss at epoch 4.

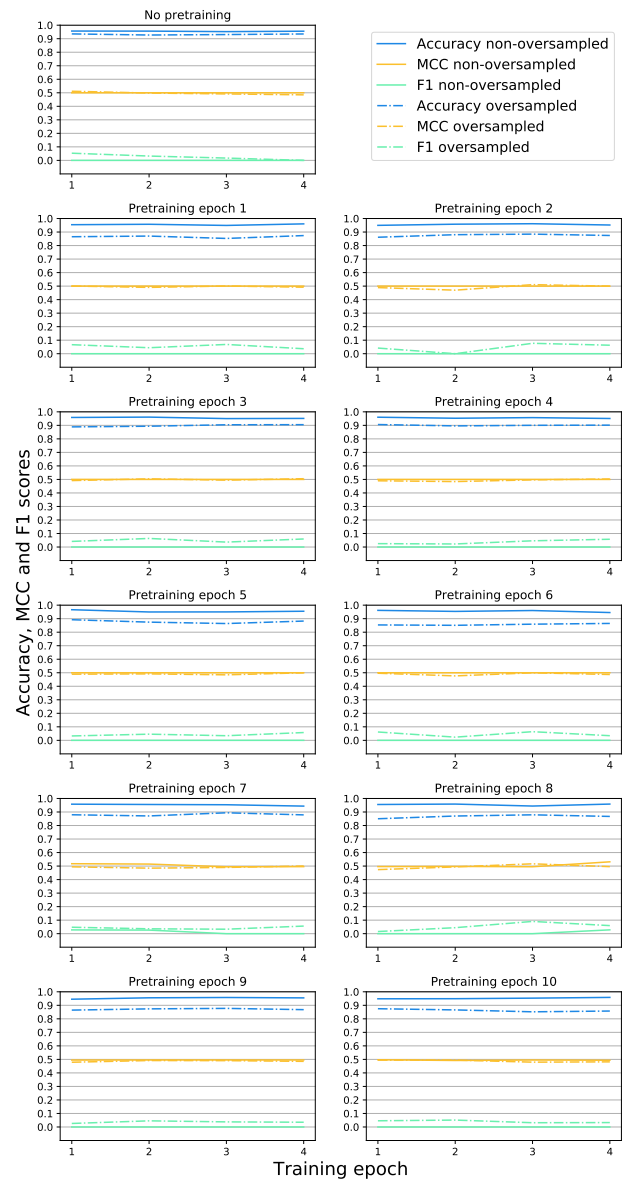


Figure 29: Validation accuracy, MCC, and F1 scores for the BERT model for empathy classification over training epochs, for every epoch of model pretraining. Transformer layers were frozen after pretraining.

Classification of a Call For Empathy in Child Help Forum Messages.

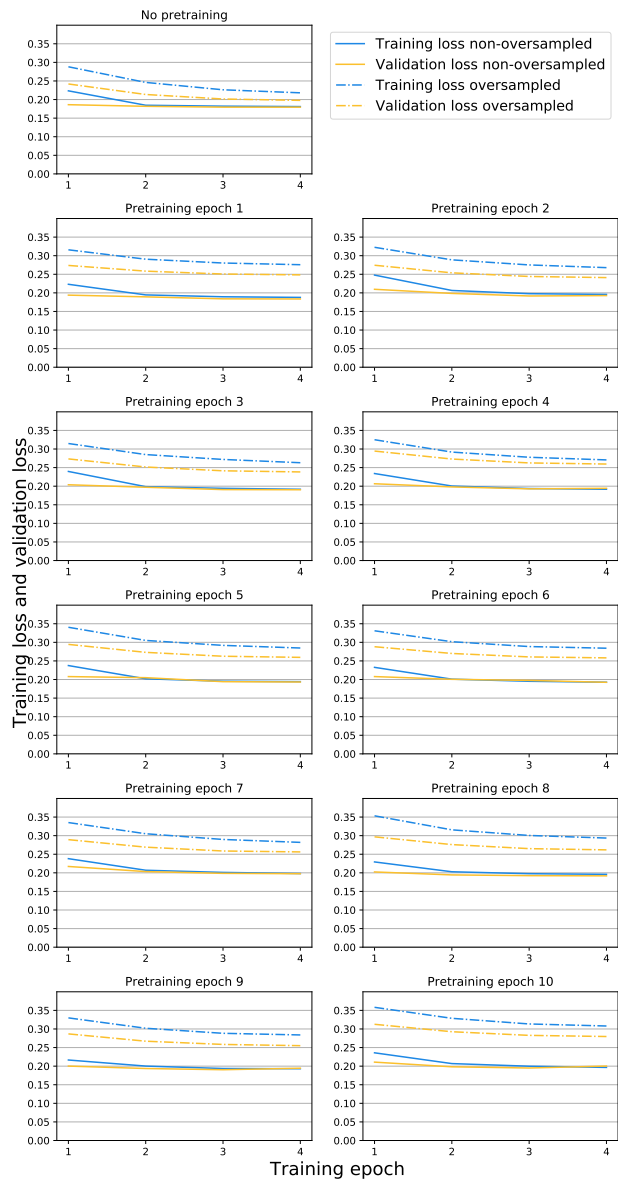


Figure 30: Model losses for the BERT model for empathy classification over training epochs, for every epoch of model pretraining. Transformer layers were frozen after pretraining.

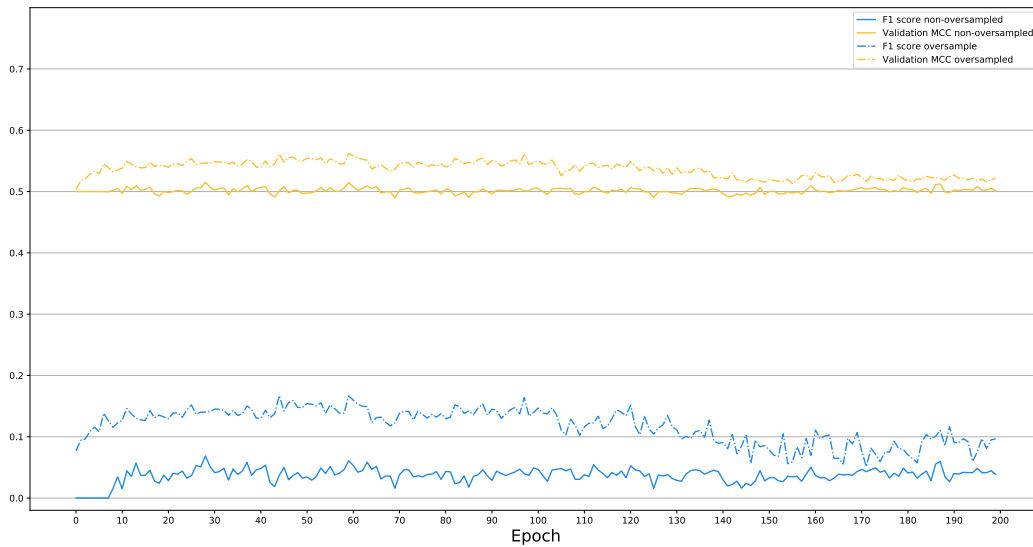


Figure 31: MCC and F1 scores for the LSTM model predicting presence of empathy over training epochs, without oversampling and with oversampling.

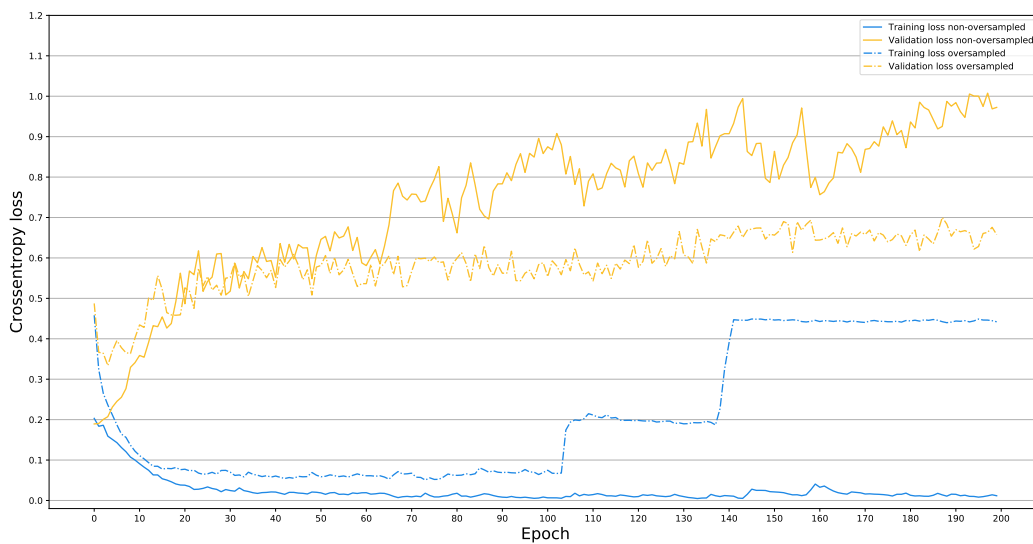


Figure 32: Model losses for the LSTM model predicting presence of empathy over training epochs, without oversampling and with oversampling.

6.4.2 LSTM

Like the BERT models, the LSTM models were trained in a five-fold crossvalidation setup, the results displayed here are a mean of the results from the five folds. Figure 31 shows the performance metrics for the LSTM model based on both Khanpour et al. (2017) and Saeidi et al. (2016), with the accompanying loss values depicted in figure 32.

Combined model. The normalized MCC score for the NOS LSTM model is nearly constant with a value of 0.5 over the 200 training epochs. The F1 score starts out at 0 for the first 7 epochs and then rises to 0.05, around which it remains.

The training loss for the NOS LSTM model starts out at 0.2 and steadily decreases over the course of the first 25 epochs

and decreases slowly thereafter, eventually hovering around 0.01. The validation loss rapidly increases from 0.2, for 20 epochs after which it increases more slowly and with more noise.

The normalized MCC score for the OS LSTM model increases from 0.5 for the first 5 epochs to around 0.55 around which it remains until epoch 100 after which it slowly decreases towards 0.5. The F1 score increases from 0.08 to 0.15 where it remains until epoch 100 after which it slowly declines.

The training loss for the OS LSTM model starts out at 0.49, decreases rapidly over the first 10 epochs to remain constant around 0.08. It sharply increases after epoch 103 to 0.2 and again sharply increases at epoch 138 to 0.45. The validation loss starts out at 0.49, decreases rapidly for 5 epochs to 0.35, then increases

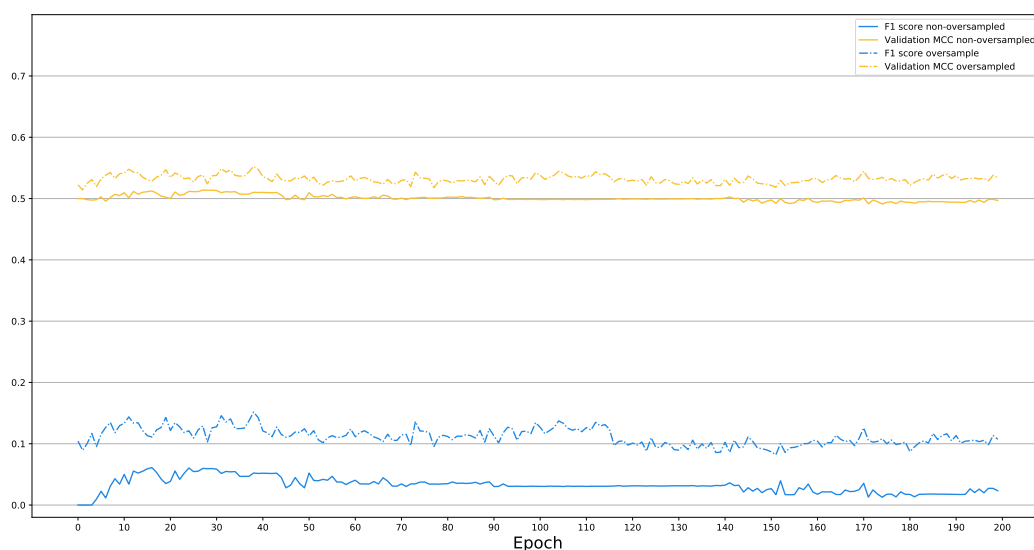


Figure 33: MCC and F1 scores for the Saeidi et al. (2016) model, predicting presence of empathy over training epochs, without oversampling and with oversampling.

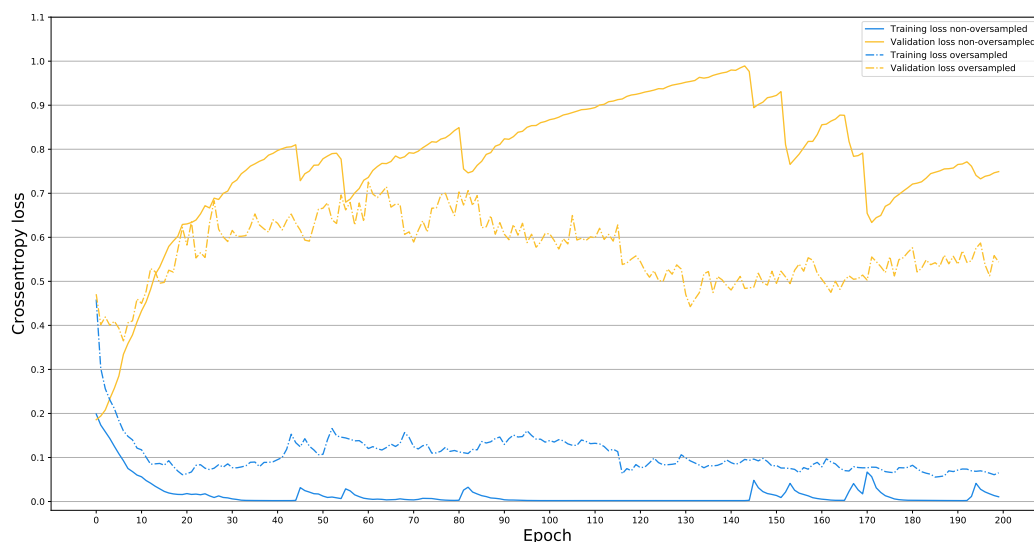


Figure 34: Model losses for the Saeidi et al. (2016) model predicting presence of empathy over training epochs, without oversampling and with oversampling.

in value, hovering around 0.55. After epoch 120, the loss increases until it reaches 0.65 around which it remains.

Saeidi model. The performance metrics for the LSTM model as implemented in Saeidi et al. (2016) are depicted in figure 5 with figure 34 displaying the accompanying loss values. The MCC score for the NOS model is mostly constant around 0.5. The F1 score increases from 0 after three epochs of training, reaching a maximum of 0.6 at epoch 15 after which it slowly decreases.

The training loss starts out at 0.2 and steadily declines until it is nearly zero at epoch 30. The validation loss increases rapidly from 0.2 without first declining. It settles at 0.8 at epoch 40 after which it shows an upward trend followed by a sharp decrease a number of times.

The MCC score for the OS model increases slightly in the first 10 epochs from 0.52 to 0.55 and slowly declines afterwards. In the same timeframe, the F1 score increases from 0.1 to 0.14, after which it decreases slightly to remain around 0.1.

The training loss for the OS model starts at 0.46 and very sharply decreases until it reaches 0.08 at epoch 20. After this, the loss increases slightly to 0.15 and decreases again after epoch 115. The validation loss decreases slightly for eight epochs, after which it increases to 0.65 which it reaches at epoch 25. It hovers around 0.65 until epoch 90 after which it drops to 0.5.

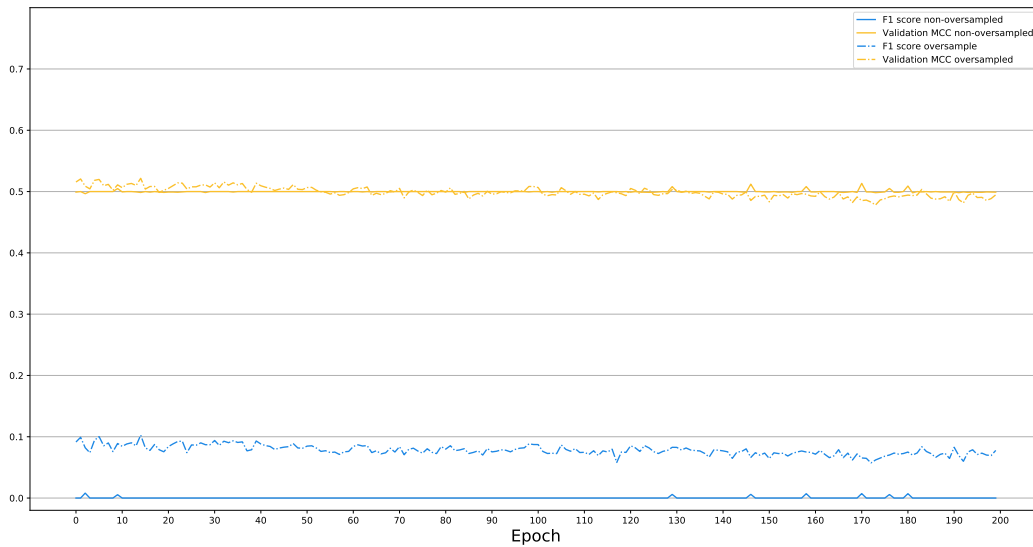


Figure 35: MCC and F1 scores for the Khanpour et al. (2017) model, predicting presence of empathy over training epochs, without oversampling and with oversampling.

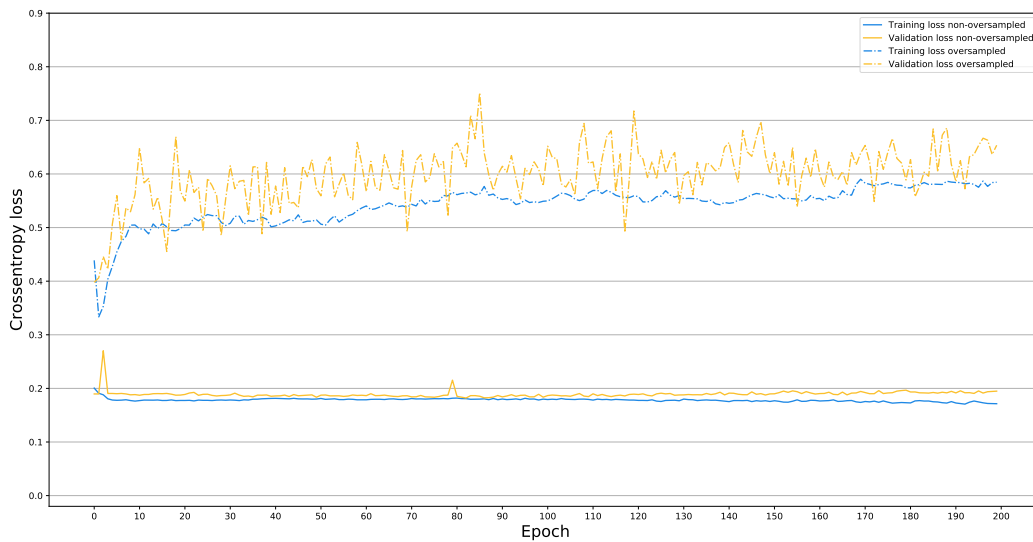


Figure 36: Model losses for the Khanpour et al. (2017) model, predicting presence of empathy over training epochs, without oversampling and with oversampling.

Khanpour model. The performance metrics for the LSTM model as implemented in Khanpour et al. (2017) are depicted in figure 4 with figure 36 displaying the accompanying loss values.

The MCC and F1 scores for the NOS models are nearly constant with a value of 0.5 and 0.0 respectively. Both the training and validation loss are nearly constant as well, with a value of around 0.2.

The MCC score of the OS model starts out at 0.51 and over all epochs slowly decreases to 0.49. The F1 score decreases very slightly over the course of the training epochs, starting out at 0.09 and decreasing to 0.08.

The training loss starts at decreases somewhat for the first three epochs and increases after this point. It reaches a value of 0.5 at epoch 10 and remains at this value until epoch 50, after

which it increases towards 0.6. The validation loss starts out at 0.43 and decreases slightly in epoch 1 but increases after this point to hover noisily around 0.6.

6.5 Call for empathy detection

6.5.1 Empathy and reply relation

The primary method for classifying call for empathy messages in this thesis is the classification of empathetic responses and the subsequent identification of the messages to which the empathetic message is a response through modelling the reply relations. The best performing models for both the LSTM and BERT architecture are tested, which are the OS BERT model based on

one epoch of pretraining and one epoch of finetuning, and the OS LSTM model after 5 epochs of training.

The BERT and LSTM models were tested along a number of thresholds for how many replies should be empathetic to label a given message as calling for empathy. All of the threshold values for the BERT model yield an MCC 0.5 or marginally over 0.5. The threshold value of more than 0 and more than 1 scored 0.508 and 0.509. For the LSTM model, none of the threshold values yielded an MCC score significantly over 0.5. Table 9 shows the full list of MCC values per model per threshold.

	LSTM	BERT
>0	0.503	0.508
>1	0.506	0.509
>2	0.51	0.504
>3	0.4985	0.5
>4	0.50	0.5

Table 9: Normalized MCC values for call for empathy classification through reply relation for different thresholds of counts of empathetic replies.

6.5.2 BERT with trainable transformer layers

As the overall performance of the reply relation algorithm was poor, the messages found to call for empathy through empathy detection and reply relation were not used to assess the direct call for empathy classification models. Instead, only the annotated call for empathy labels were used.

Figure 37 shows validation accuracy and normalized MCC values for the four training epochs for all ten pretrain models and for a model without pretraining. The graphs show the NOS models as well as the OS models. All models were trained in a fivefold crossvalidation scheme, the graphs shown are averages of the five folds. These graphs show models in which both the transformer layers as well as the classification head are trained.

The NOS BERT model for direct call for empathy detection shows a somewhat similar performance across all pretraining base model variations. The accuracy scores are nearly constant with a value of 0.98. The MCC scores vary between 0.90 and 0.95 and show no clear trend upward or downward across all conditions. The F1 scores follow the trend of the MCC scores, though again lower, varying between 0.78 and 0.86.

The loss values for the NOS BERT models all show the same general trend. The training loss declines steadily throughout the epochs, flattening out somewhat between epochs 3 and 4. The validation loss starts out lower than the training loss and remains constant between epochs 1 and 2, rises slightly between epochs 2 and 3, crossing the training loss line, and remains mostly constant between epochs 3 and 4.

The accuracy of the OS models is notably lower than then NOS models with a nearly constant value of 0.9. The MCC scores and F1 scores for the OS BERT models quite close to the NOS models' equivalents but are generally slightly lower. As was the case for the empathy prediction models, the models based on one epoch of pretraining did not perform better than the models based on no pretraining for either NOS or OS conditions.

The loss values for the OS BERT models are all fairly similar. The training loss steadily decreases over all four training epochs. The validation loss increases very slightly over the first two epochs, except in pretrain conditions 4 and 5 in which the validation loss increases more sharply. After epoch 3, the validation

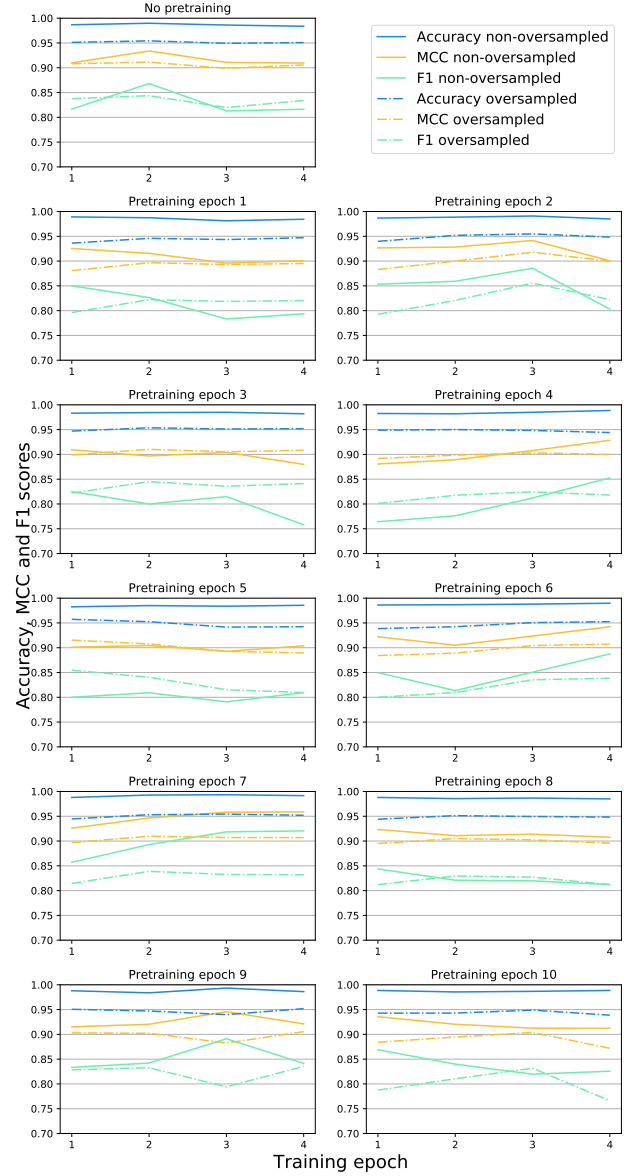


Figure 37: Validation accuracy, MCC, and F1 scores for the BERT model predicting the call for empathy over training epochs, for every epoch of model pretraining.

loss increases somewhat more sharply in all pretrain conditions. The graphs of the loss values of both the NOS and OS models are shown in figure 38.

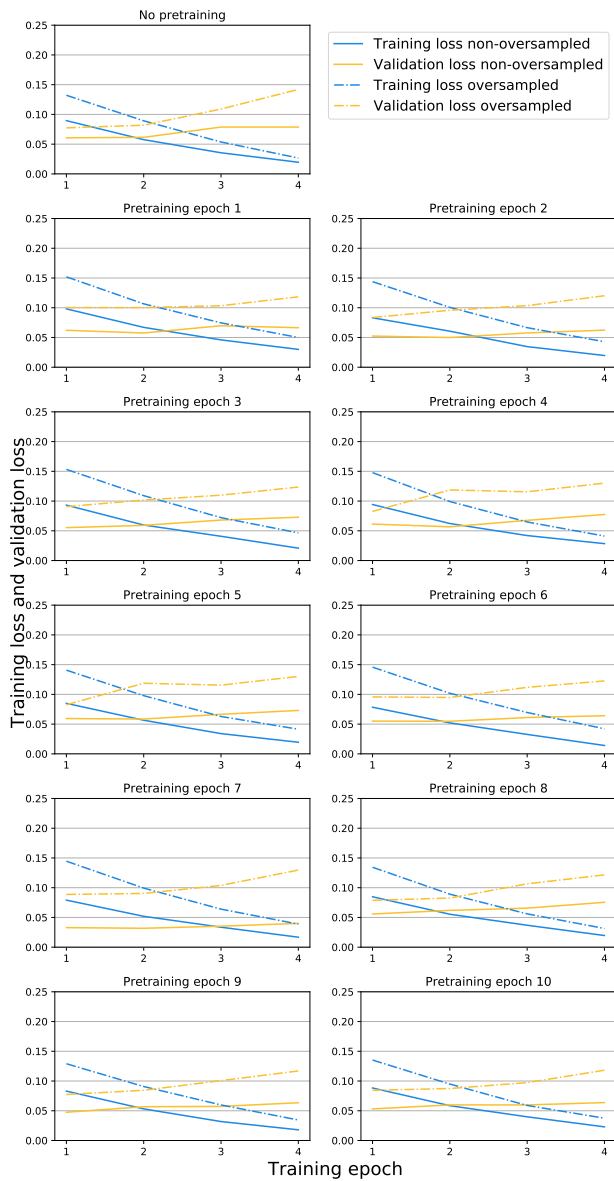


Figure 38: Model losses for the BERT model predicting the call for empathy over training epochs, for every epoch of model pretraining.

6.5.3 BERT with frozen transformer layers

As was the case for the empathy classification BERT models, the call for empathy classification models were also run with the transformer layers frozen after pretraining. Only the classification heads are trained in these models. Figure 39 shows the performance metrics for these models, and figure 40 shows the associated losses.

The NOS models show a mostly constant accuracy with a value of around 0.85, MCC scores almost exactly 0.5 and F1 scores of

almost exactly 0 over the course of the four epochs. The accuracy, MCC scores and F1 scores for the models trained on oversampled data are all similar and fairly constant, with a value of 0.9 for the model based on no pretraining and a slightly lower value of 0.85 for the other pretrain conditions.

The training and validation loss values for the both the NOS and OS models are all very similar. They all either decrease very slightly after the first epoch of training and remain constant thereafter or remain constant throughout the training epochs. The loss values of the OS models are generally slightly lower than the loss values for the NOS models.

Classification of a Call For Empathy in Child Help Forum Messages.

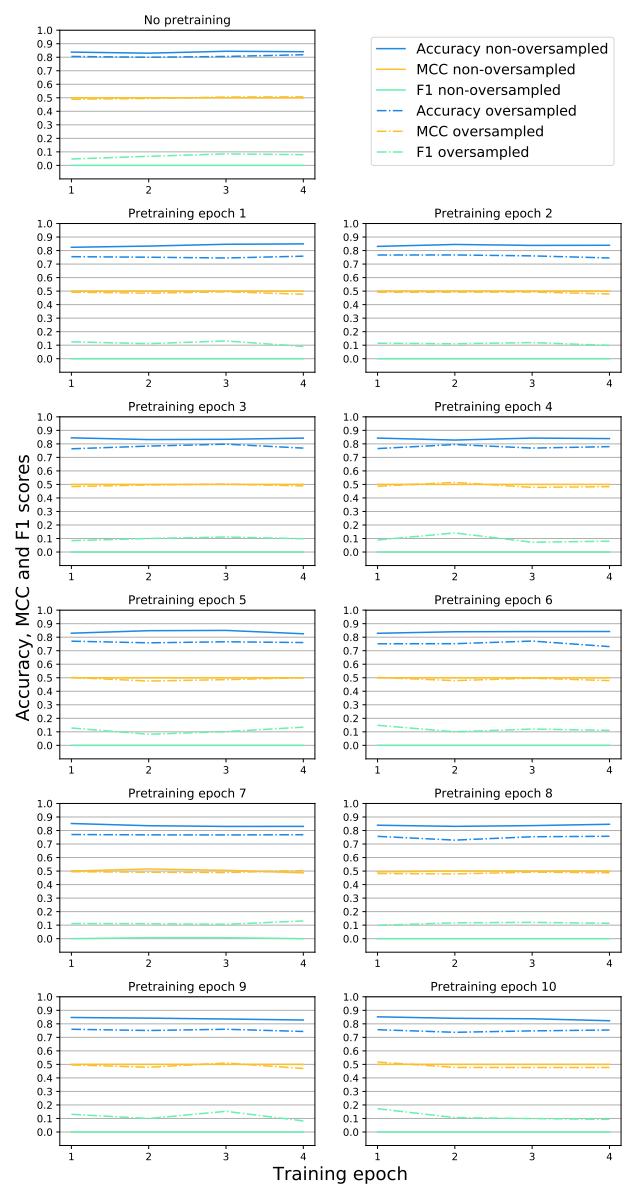


Figure 39: Validation accuracy, MCC, and F1 scores for the BERT model predicting the call for empathy over training epochs, for every epoch of model pretraining. Transformer layers were frozen after pretraining.

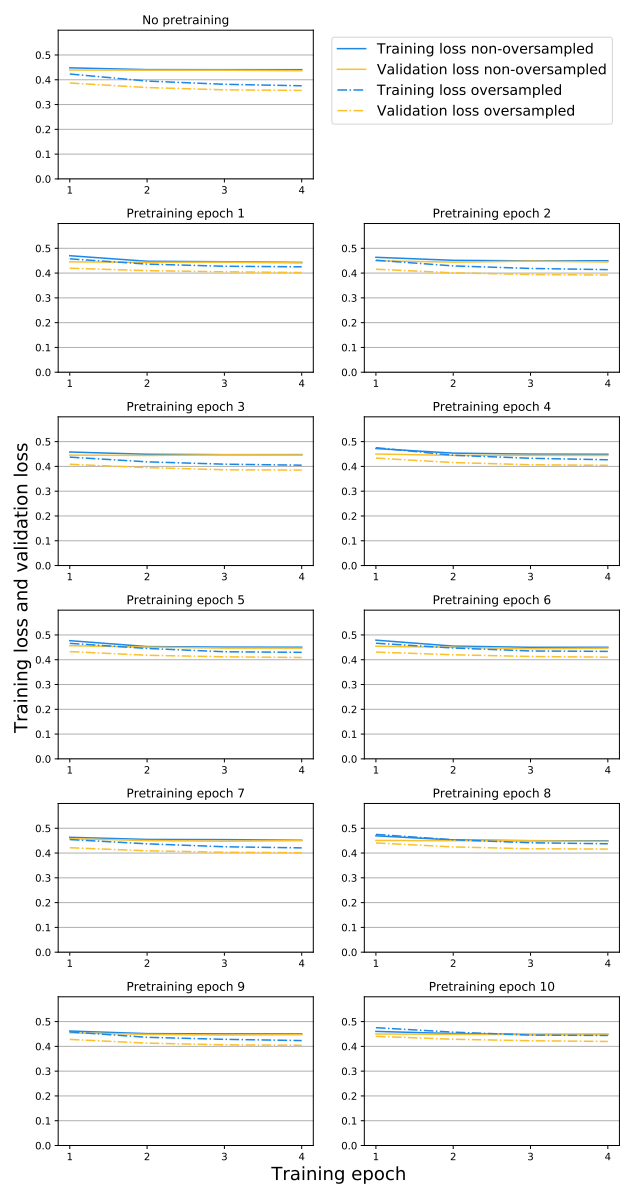


Figure 40: Model losses for the BERT model predicting the call for empathy over training epochs, for every epoch of model pretraining. Transformer layers were frozen after pretraining.

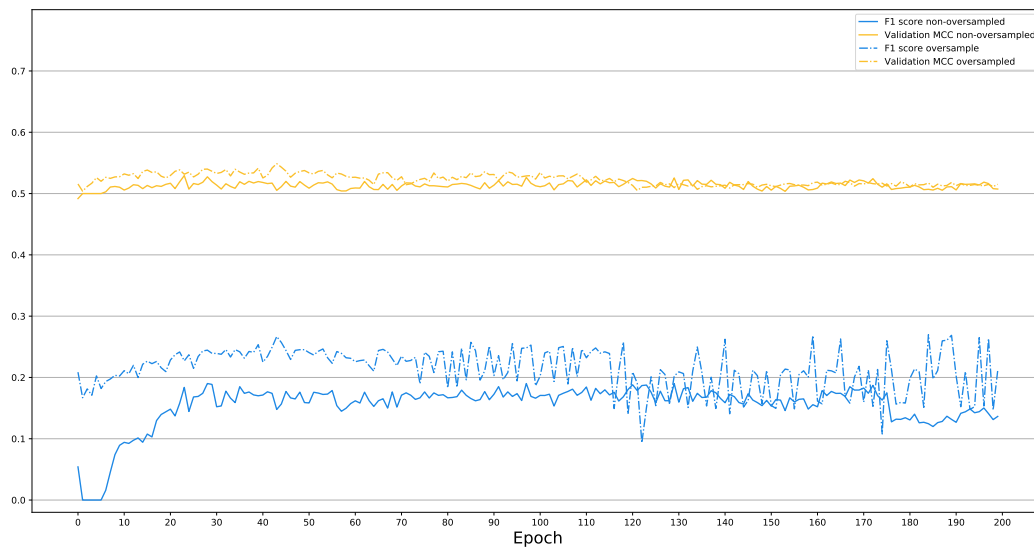


Figure 41: MCC and F1 scores for the LSTM model predicting call for empathy over training epochs, without oversampling and with oversampling.

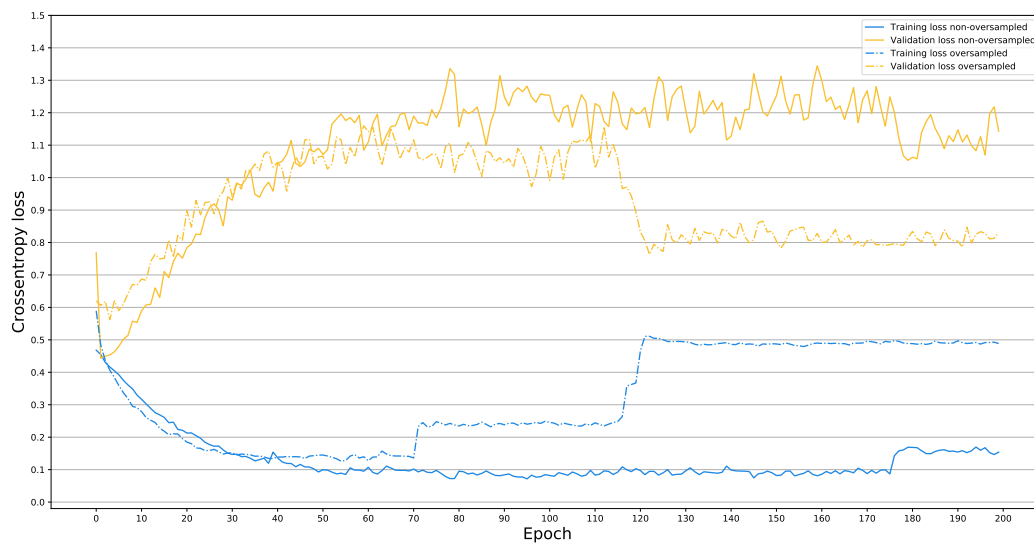


Figure 42: Model losses for the LSTM model predicting call for empathy over training epochs, without oversampling (upper) and with oversampling (lower).

6.5.4 LSTM

Combined model. Figure 41 shows the performance metrics for the LSTM model based on both Khanpour et al. (2017) and Saeidi et al. (2016), with the accompanying loss values depicted in figure 42.

The MCC score for the NOS LSTM model starts out 0.5 and increases slightly to 0.51 at epoch 10, after which it remains mostly constant. The F1 score decreases from 0.05 to 0 after epoch 1 and increases again after epoch 5, reaching 0.19 at epoch 28. After this point, the F1 score remains mostly constant around 0.18 until epoch 175, when it drops to 0.12.

The training loss of the NOS LSTM model declines from around 0.45 to 0.1 over the course of 50 epochs of training, after which

it remains constant until epoch 175, at which point it increases sharply to 0.15. The validation loss decreases sharply the first three epochs, after which it increases steadily to 1.2 during 50 epochs, where it remains mostly constant with some noise.

The MCC score of the OS LSTM model increases from 0.5 to 0.53 after 5 epochs of training, after which it slowly declines back to 0.5. The F1 score increases from 0.18 to a maximum of 0.25 at epoch 43, after which it declines and becomes increasingly noisy.

The training loss of the OS LSTM model decreases sharply at first from around 0.58, but flattening out at epoch 30. At epoch 70, the loss suddenly increases a little after which it remains mostly constant until epoch 115 at which point it increases sharply to 0.5 after which it remains constant. The validation loss very

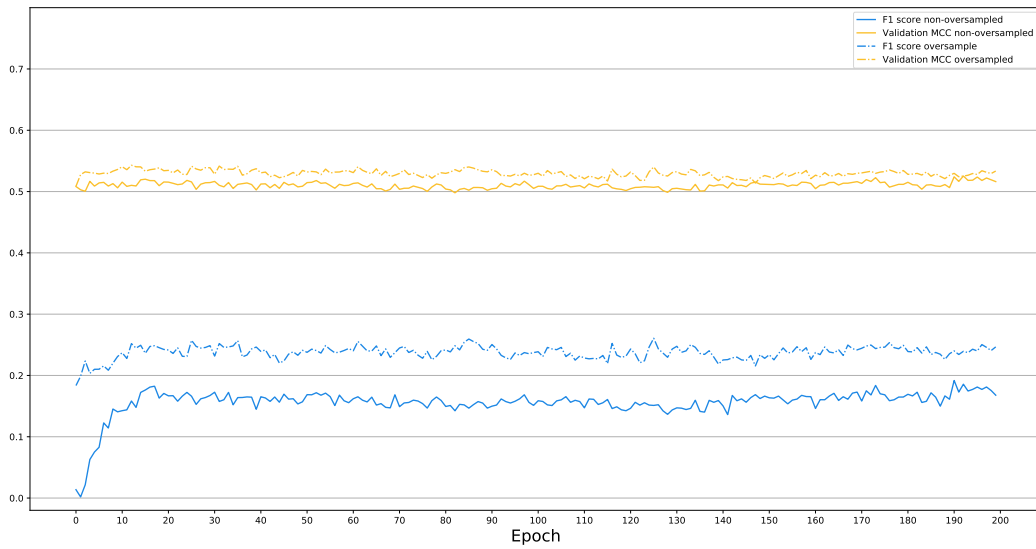


Figure 43: MCC and F1 scores for the Saeidi et al. (2016) model, predicting presence of call for empathy over training epochs, without oversampling and with oversampling.

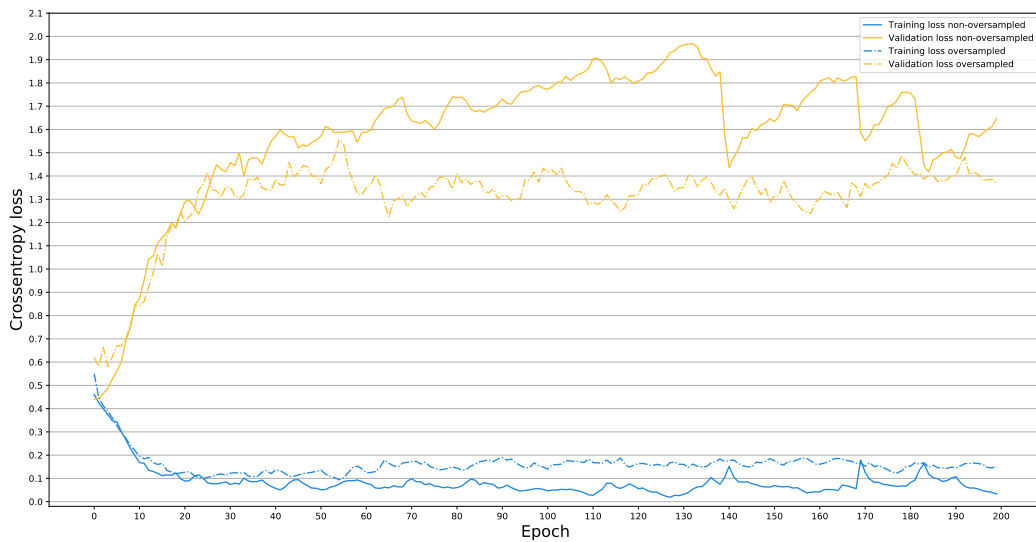


Figure 44: Model losses for the Saeidi et al. (2016) model, predicting presence of call for empathy over training epochs, without oversampling and with oversampling.

slightly decreases for the first four epochs but increases steadily afterwards until epoch 40, after which it remains mostly constant until epoch 110, where it suddenly drops to 0.8 to remain around this value.

Saeidi model. Figure 43 shows the performance metrics for the LSTM model as implemented in Saeidi et al. (2016), with the accompanying loss values depicted in figure 44.

The MCC score for the NOS Saeidi model is mostly constant and remains around 0.51. The F1 score increases sharply to 0.18 at epoch 15, after which it remains mostly constant at 0.17. The training loss decreases sharply to 0.1 at epoch 10 after which it declines slowly towards 0.5. The validation loss increases sharply without first decreasing from 0.45 to 1.5 over the first 30 training epochs, after which it increases with a shallower incline. The

validation loss suddenly drops a number of times after which it slowly increases again.

The MCC score of the OS Saeidi model increases from 0.5 to 0.53 after two epochs of training and remains constant at this value. The F1 score increases from 0.19 to 0.25 over 12 epochs of training and remains constant at this value. The training loss decreases sharply from 0.45 towards 0.1 for 15 epochs, after which it slowly increases towards 0.2 over the remaining training epochs. The validation loss remains mostly constant for the first eight epochs and increases sharply after this point, hovering around 1.35 after 25 epochs.

t

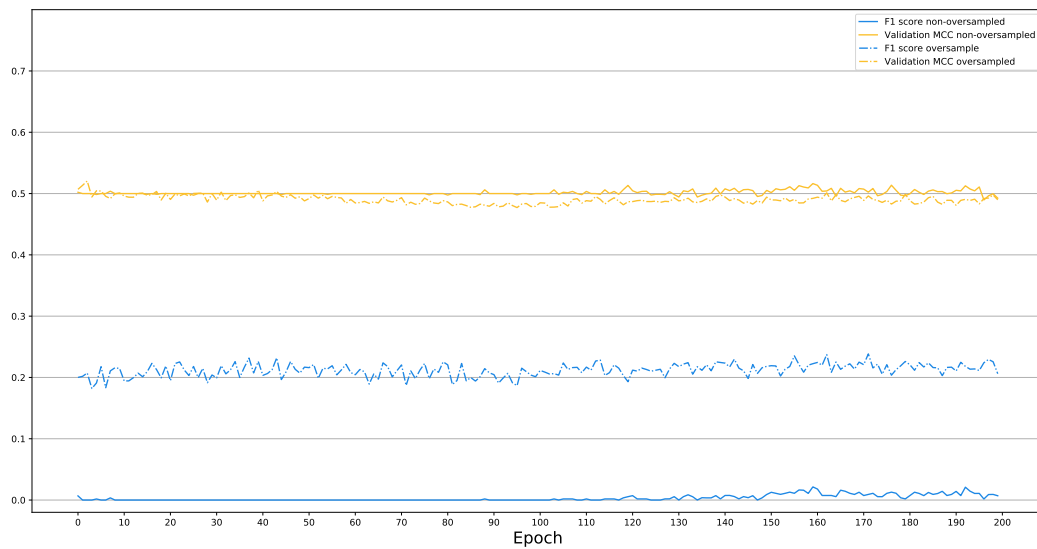


Figure 45: MCC and F1 scores for the Khanpour et al. (2017) model, predicting presence of call for empathy over training epochs, without oversampling and with oversampling.

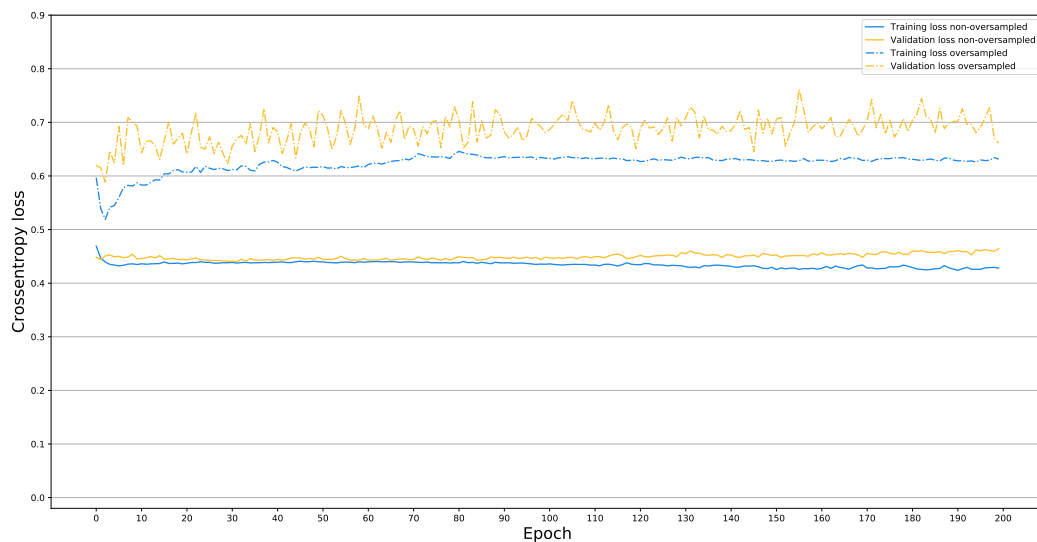


Figure 46: Model losses for the Khanpour et al. (2017) model, predicting presence of call for empathy over training epochs, without oversampling and with oversampling.

Khanpour model. Figure 45 shows the performance metrics for the LSTM model as implemented in Khanpour et al. (2017), with the accompanying loss values depicted in figure 46.

The MCC and F1 scores for the NOS Khanpour model are mostly constant, with values of 0.5 and 0.0 respectively. The training loss decreases slightly over the first three training epochs but remains constant afterwards. The validation loss remains constant throughout the training epochs.

The MCC value of the OS Khanpour model starts out around a value of 0.5 and decreases to 0.49 over the course of the training epochs. The F1 score increases very slightly over the course of the training epochs from 0.2 to 0.22. The training loss decreases slightly the first 4 epochs and increases after this point, reaching a value of 0.6 at epoch 20. It increases slowly until it remains

constant at a value of 0.62 after epoch 90. The validation loss decreases very slightly during the first three epochs but increases thereafter to hover noisily around 0.68, and increases slightly to 0.7 after epoch 100.

7 Discussion

7.1 Annotation agreement

Of the total of 71 annotators which were marked for removal by the comparison script, two annotators were kept after manual evaluation. This evaluation was not set up with exact predefined criteria. As the obviousness of the falsity of the labels was larger than expected, there was a clear distinction between annotators who mislabeled posts in an obvious and consistent manner and annotators who performed the task seriously. Even in cases where the ‘is empathy’ label was correctly given, the empathy type and empathy valence labels indicated poor annotation quality for inattentive annotators. This made a clear decision boundary for the removal or inclusion of annotators marked to be removed by the annotator evaluation script possible.

Examples of obvious false labels are given in figure 47, which shows two example messages with empathy annotations given by poor annotators. The two annotators which were kept in the dataset despite being marked as poor annotators lacked these obvious false labels entirely.

7.2 Reply relations

The majority of the labeled reply relations were labeled by the proximity based similarity component. This imbalance of the proportions of the components suggests that the inclusion criteria for most of the components are too strict. This might have different causes for different components. The Z-scores indicate that none of the components perform significantly different from chance level for that component. This means that the assumptions underlying the several components may be false, that the selection criteria for a component were poor or that the annotations were wrong.

The list type component covers far fewer messages than expected with only 4%, which might be caused by poor pattern matching. There might be patterns in the lists that were not detected by the filter, such as lists that start with bulletpoints (●), asterisks (*) or other indicators. The assumption that most of the messages are a response to the original post is not likely to be flawed. The inclusion criteria potentially misses list threads, as described above, but may also include posts which are not in fact lists but contain an enumeration or list, or very small sentences with many linebreaks. This would apply a far too simple paradigm on threads which are more dynamic than the list threads, which leads to poor results.

For both the actual mention and implied mention component, the assumption that mentions always refer to the last post in the same thread made by that user was made. While this may often be true, it is not true for all mentions. For example, a message containing the sentence fragment “... just like *BlueJeans* said, I think it is...” does not reply to the last message made by user *BlueJeans* but rather mentions it as a support for their argument. The lack of distinction between these two uses of mentioning might contribute to the poor performance of the direct mention component. A distinction could be made by including the placement of the mention. A mention in the middle of the text as was made in the example seems to refer to another user in passing, while a mention at the start of a post seems more likely to indicate a reply relation. This distinction can be verified using the reply relation annotations collected in this study and may improve the reply relation resolving algorithm used.

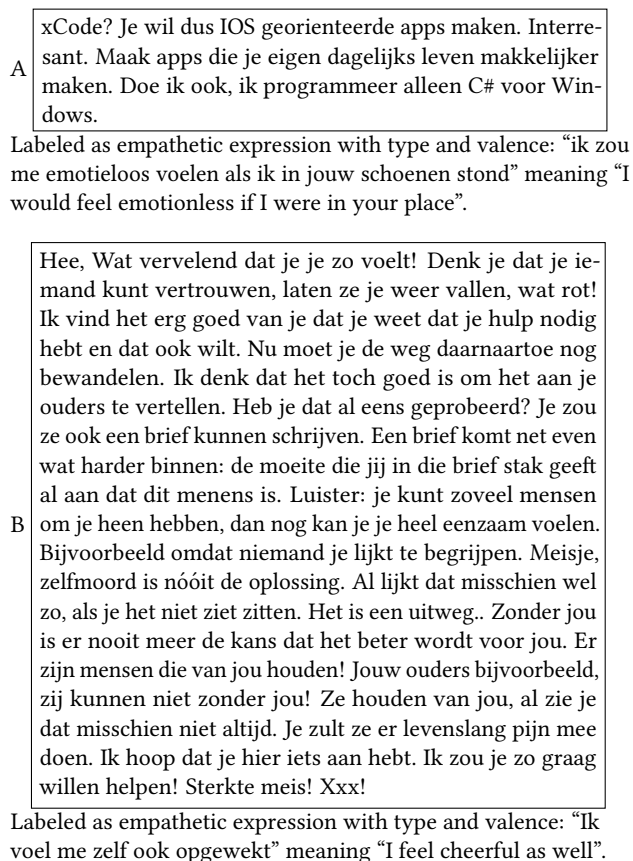


Figure 47: Example of an annotation so clearly mislabeled that it is indicative of lack of effort from the annotator (A) and an example of an annotation in which the ‘is empathy’ label is correct but type and valence are still clearly mislabeled. (B)

The similarity component was meant to detect post which literally copied a previous post and appended an answer and thus the threshold for similarity was set quite high. If this inclusion criterion is realistic, the proportion of 0.05 implies that these types of mention occur with a frequency of around 5% of all posts, which seems plausible. However, it is also possible that the replies added to the copied text were longer than expected. If the replies are indeed longer than expected, a sliding window comparison would yield better results than an overlap comparison of the entire text. This sliding window would include messages in the component if a section of some specified length is exactly the same as another message, and would be agnostic to how long a reply is. For this component, the poor performance in reply prediction might lie in the assumption of the most similar previous post being the antecedent. This assumption is only true for the first post which directly quotes another post, but if subsequent posters also quote the same earlier post but respond to the first quoter, a patterns which occurs somewhat frequently on the Kindertelefoon forum, the component selects the wrong antecedent.

The implied mention component was not used once, which is lower than expected. As was the case for previous components, the inclusion filter might have been too strict. It might be the case that implied mentions slightly deviate from the actual username,

and for example user *BlueJeans001* might be referred to as *blue-jeans*, which would not pass the implied mention filter. A good improvement over the currently used exact matching would be a minimal edit distance matching mechanism. One concern for the implied mention was that users might be mentioned accidentally, especially if a user has a name which is also a common phrase or word. However, for this to have happened, the user also needed to have responded to the same thread previously, which lowered the chances considerably. The count value of 0 indicates that this did not happen but this might be a concern if a minimal edit distance filter was employed with too lenient parameters.

The number of question answer pairs is also much lower than expected at 1%. The question answer pair component only yields an antecedent if a question is detected by the question detection algorithm and when there is a suitable answer found. If a question is detected but there is no similar enough post which may be an answer, there is no question-answer pair formed. This can be made more inclusive by testing the question detection algorithm against a number of thresholds and picking a less conservative but still properly functioning threshold.

The question classification algorithm was the only component for which a label was included in the annotation, so it can be evaluated in more detail. While the accuracy of 0.7 for the chosen question detection algorithm seems decent, the MCC score reveals that this is due to the skewed data the algorithm is trained on. The MCC score indicates that the question detection algorithm does not score much better than chance level, taking into account the skewed prior probabilities of the ‘is answer’ label. The effect of this poor performance on the whole is quite low however, as the question answering component only contributed 1% of the found antecedents.

The proximity weighted similarity component contributed by far the most of the reply relation algorithms used, with 88% of the antecedents. As this was the last component in the decision tree, the best estimate was used and there were no inclusion criteria. The implementation of the proximity weights was such that it had a very low impact on short threads and only a moderate impact on longer threads. This was done to prevent long reply relations for being overlooked but resulted in a very low proximity based weight, which resulted in the closest matching message being selected as antecedent regardless of proximity in most cases.

7.3 BERT pretraining

The decrease in validation loss from no pretraining on the Kindertelefoon data to epoch 1 of pretraining indicates that one epoch of pretraining on representative data increases the performance on the pretraining tasks significantly. This increased task performance is reflected in this accuracy shown in figure 26, which shows a large increase in accuracy between epoch 0 (no additional pretraining) and epoch 1.

The increase in validation loss and the decrease in training loss starting from epoch 1 of pretraining is indicative of overfitting the training data. However, the accuracy scores in figure 26 do not indicate this, as both the training and validation accuracy remain almost constant after epoch 1, where the training accuracy was expected to go up and validation accuracy to go down if the model was overfitting the training data.

One explanation for the constant accuracy scores and increasing validation loss is that outliers are predicted increasingly poorly. Since the outliers did not contribute to the accuracy score regardless of how poorly they were predicted, the accuracy

score is not expected to change. However, because of the logarithmic nature of the loss function used, the increased loss for a small number of items can affect total loss score significantly.

Another explanation is for this behaviour is that the prediction probability for all datapoints decreases by a similar amount, which results in the same accuracy, as the highest probability is chosen. Since the training and validation data are randomly determined every epoch, it is difficult to track individual predictions through the epochs hence this behaviour cannot be fully explained. Regardless, the increase in validation loss without an increase in accuracy implies that the model is no longer learning new useful patterns from the data.

Because the significant increase in pretraining tasks performance is an indication of natural language understanding, the model performance of a model based on one epoch of training data was expected to be significantly better than a model based on the Bertje model with no additional pretraining on Kindertelefoon data. As the model does not improve after the first epoch, models based on subsequent pretraining epochs were not expected to be a significant improvement over the model based on the first epoch of pretraining.

As the BERT model used in this study was based on the Bertje model (Vries et al., 2019), the Bertje vocabulary was used. This vocabulary is based on Dutch books, Dutch news and the Dutch wikipedia, which does not use the same vocabulary as is used on the Kindertelefoon forum. However, because of the wordpiece word representation, many words could be interpreted in word-piece form. Unparsable words were mostly (slang) abbreviations, which have a low impact on empathy classification performance. The large increment in pretrain task performance indicates that the BERT model is able to perform natural language understanding tasks and hence is not impeded by the unparsable words in the Kindertelefoon vocabulary to an extent that it performs poorly.

7.4 Empathy classification: BERT

7.4.1 Result interpretation

Since the messages labeled as empathetic constitute only 5% of the documents in the dataset, the high accuracy scores on all of the BERT models do not constitute strong evidence that the minority class as well as the majority class are being predicted correctly. The normalized MCC scores of between 0.87 to 0.93 for the NOS empathy classification models are well above 0.5, the point in the normalized MCC scale indicating chance level predictions, indicating a strong predictive power.

The MCC scores for the OS empathy detection models were similar to the NOS models, though they were in general slightly higher. This was most noticeable in the models based on no pretraining and one epoch of pretraining. This was expected, as oversampling should give the model more examples to train on, improving its ability to distinguish between the classes. A performance improvement would indicate that the oversampling method was successful in creating messages similar to the minority class messages they were based on, but the observed improvement observed is too small to conclude this.

The loss values of both the NOS and OS BERT models as depicted in figures 28 show a training loss which decreases and a validation loss which increases. This could be an indication of the model overfitting the data, however the increase in validation loss is small for these models, so it is not a concern within the

four epochs each model is trained. The fact that the validation loss does not decrease with the training loss does mean that there are no new patterns learned which are useful for decreasing loss on new data, which means that there is likely no gain to be had in longer training.

Taking these loss values into consideration in combination with the performance scores, the model yielded from epoch one with one epoch of pretraining as basis performed best for both the NOS and OS models.

Pretraining performance gain Based on the pretraining tasks, the BERT model for empathy classification based on one epoch of pretraining was expected to perform better than the model based on no pretraining, and no improvement was expected from models based on subsequent pretraining epochs over the model based on the first training epoch. The MCC scores of the OS BERT models indicate that the model based on one epoch of pretraining does perform slightly better than the model based on no pretraining but this difference is not present in the NOS models. Furthermore, the difference between the MCC scores of the model based on no pretraining and the model based on one epoch of pretraining is smaller than expected. As was expected, the models based on more than one epoch of pretraining did not outperform the model based on no pretraining nor the model based on one epoch of pretraining.

The absence of a significant performance gain might be due to the fact that during the task training, all layers were updated, not only the last layers. As this trains the transformer layers as well, the effect of pretraining can be achieved during training on the empathy detection task. This would give the models based on no pretraining a similar training to one epoch of pretraining, and the equivalent of two epochs of pretraining for the model based on one epoch of pretraining. Because performance did not increase in the pretraining tasks after one epoch, it makes sense that the performance in the model based on one epoch of training on the classification task does not perform better than the model based on no pretraining. Because of this consideration, all models were also tested with the transformer layers frozen. Figure 29 in section 6.4.1 shows the performance metrics for these models, with the associated losses depicted in figure 30.

Frozen transformer layers The NOS models trained with frozen transformer layers show a constant MCC score of 0.5 and F1 score of 0, indicating a chance level performance. The accuracy scores are fairly high because of the imbalance of the data and because the models did learn the prior probability of the classes and as such predict the majority class most of the time. The OS models do not show significantly better performance, even though the F1 score is slightly higher than zero. The model losses for both the NOS and OS models show that there is little improvement over the four epochs, with only a very slight decrease in loss after the first epoch.

The expected performance gain of the model based on one epoch of pretraining over the model based on no pretraining was absent in the models with frozen transformer layers, as they all performed at chance level.

This poor performance indicates that the classification process in the BERT model does not consist of a content encoding in the transformer layers and a classification in the classification head. Rather, the classification task is trained into the transformer layers of the model and the transformer layers capture task-specific patterns in the data aside from more generalizable natural language understanding of the text. It also indicates that the

natural language understanding concerning the Kindertelefoon data as trained on the masked language model task and next sentence prediction task does not suffice to encode the input data in such a way that the classification head alone can train well on the empathy and call for empathy classification tasks.

7.5 Empathy classification: LSTM

The LSTM model developed in this study, which is a combination of the models implemented by Khanpour et al. (2017) and Saeidi et al. (2016), was tested as well as the separate models by Khanpour et al. (2017) and Saeidi et al. (2016).

The combined model and the Saeidi et al. (2016) model show many similarities in performance and loss metrics. The difference between NOS and OS models is fairly small and neither model architecture in either NOS or OS condition performs significantly better than chance level. The MCC score of the NOS models does not deviate from the 0.5 line, and while the F1 score does increase from zero after a small number of training epochs, the peak F1 score is not very high. This indicates that the models did not predict both classes much better than chance. For these models, the validation loss only goes up, while the training loss goes down. This indicates that the models overfit the training data, even though they do not at any point fit the data in general well enough to decrease validation loss.

The OS models show a small increase in MCC which was nonetheless only marginally higher than chance level. These models show a clearer point at which the training should be stopped. The validation loss is lowest after epoch 5 for both the combined model and the Saeidi et al. model. After this point, the validation loss rises and models overfit the training data. For a minimal loss, training should be stopped at 5 epochs, which coincides with near peak performance. Unlike the NOS models, the OS models have trained well enough to decrease the validation loss and increase the MCC score at least to a small extent.

The Khanpour et al. model in the NOS condition does perform better than chance level. Furthermore, the flat loss values indicate that the model failed to capture patterns in the training data. The convolutional layers, which act as trainable filters for data patterns in this context, failed to capture the higher level patterns of patterns which they were designed to do.

The F1 score for the OS Khanpour et al. model does deviate from zero, but does not increase over training epochs. Since the MCC score does not deviate from 0.5, the predictive power of the model for both classes is no better than chance level. The small decrease in training loss implies that there was some pattern that was learned by the OS model that was not learned by the NOS model. However, since the training loss increases after this small dip, this was not a pattern that fits the whole training data. By extend, these patterns do not describe the validation data well, which can be seen in the increasing validation loss. Despite not learning, the base F1 score for the Khanpour et al. OS model is higher than the base F1 score for the NOS model.

Neither of the reference models comes close to the performance described in the papers in which they were published. The output shape of the Saeidi et al. model was adapted to suit the data used in this study but this is not expected to influence the model performance in such a magnitude. Since this model was built and evaluated on a much simpler task, the performance difference might be caused by the model simply not being able to recognize any patterns which point to the presence of empathy.

The convolutional layers in the Khanpour et al. model architecture can act as filters to detect word patterns as sequences of three-word sequences. It is likely that the patterns defining empathy cannot be captured in this way and that the Khanpour et al. model does not work well for this reason. This might be due to the creative and less homogeneous language use on the Kindertelefoon forum than for example can be found in documents written by health professionals or adult peers who are in a similar situation as the empathy target.

7.6 Call for empathy

Both the LSTM and BERT-based reply relation mediated call for empathy predictions showed a chance level performance for all thresholds used. Since the LSTM models showed no significant predictive power in empathy classification, the lack of predictive power in call for empathy detection was expected. The BERT model, which performed significantly better in empathy classification, did not show a better predictive power, as the mediating reply relations algorithm performed poorly. As the performances compound, the poor performance of the reply relations algorithm cause the total mediated call for empathy classification to perform poorly.

The direct prediction BERT models performed better in the call for empathy classification than the reply-relation mediated approach, with MCC scores over 0.9 for both the NOS and OS models. These scores indicate a strong predictive power for both classes. The increased performance of the OS models versus the NOS models that was present in some of the BERT for empathy classification models is absent in the call for empathy BERT models. The difference between NOS and OS models is negligible.

The loss values for the NOS and OS models are similar. They show a steadily decreasing training loss and a validation loss which is constant at first and slowly increases after epoch 2. This pattern indicates that there are no new useful pattern learned by the model after epoch 1, as the validation loss does not go down after epoch 1. As the performance of the models based on no pretraining are or par with or better than the other pretraining conditions, the models from this condition are considered the best performing models. This is true for both the NOS and OS models.

The BERT models for call for empathy classification were also run with the transformer layers frozen to give an insight into the effect of pretraining on the models. Like the empathy classification BERT models, the MCC scores for both the NOS and OS models did not deviate from 0.5. The F1 scores are generally slightly higher for the OS model but still very low. The poor performance combined with the nearly constant training and validation loss values indicates that neither the NOS nor the OS models were able to learn patterns useful in classification and that the transformer layers capture task-specific patterns as well as patterns which aid a more general natural language understanding.

The NOS LSTM model developed in this study and the NOS Saeidi model show a very small increase in MCC above 0.5, which was too small to be considered of value. The F1 scores for both the combined model and the Saeidi et al. (2016) model rise up from zero which implies a greater performance increase than the MCC scores indicate, though neither architecture comes close to the performance described in (Saeidi et al., 2016).

The OS versions of these model architectures show a slightly larger MCC score gain which is nonetheless very small. The F1

scores indicate a small performance gain of the OS models with regards to the NOS models.

The OS combined LSTM model and NOS and OS Saeidi et al. model overfit on the training data from the first epoch onward. The NOS combined LSTM model shows a drop in validation loss for a small number of epochs but then also overfits the training data.

The combined OS combined LSTM model (and to a smaller extent the NOS combined LSTM model) show sudden increases in training loss and sudden drops in validation loss. This is indicative of vanishing gradients in the softmax output layer of the model.

Neither the NOS nor OS Khanpour model show any indication of pattern recognition in the training dataset. The flat training and validation loss of the NOS model indicate that it was not able to train. The loss values for the OS Khanpour model show a similar difficulty in training, although the training loss does decrease slightly at the start of the training session.

7.7 Oversampling

The models based on oversampled data (OS models) showed slightly better results in the BERT and LSTM models for empathy detection as well as in the combined and Saeidi et al. LSTM models on call for empathy classification, but not the BERT model for call for empathy classification.

The performance gain is an indication that the oversampling has successfully provided the model with more examples to train on. However, there are also disadvantages to oversampling. The OS models are not able to learn the probability distribution of the data, and predict more empathetic documents than there are in the validation set (about 50 % more) whereas NOS models are much closer to the number of empathy documents present in a given dataset. This higher empathy class prediction rate of the OS models explains the increased F1 value for the Khanpour models which does not seem to have learned any patterns useful in classification yet has a non-zero F1 score. Additionally, it likely impacts the other LSTM based models as well, especially the models predicting call for empathy as these models show a relatively large discrepancy between F1 and MCC values. The OS BERT models do not suffer from this caveat and have fairly homogeneous label counts which are all close to the proportion in the validation dataset.

The models may also be able to identify the oversampled documents if they differ in a systematic way from regular documents, which could lead to a better OS model performance. Since all oversampled documents are of the minority class, this leads to a very well defined distinction between the classes. While the oversampled texts do share common themes with the non-oversampled texts they were based, they do have a distinct lack of correct grammar, which can be seen in the example sentences in figure 22. This also means the models trained partly on nonsensical features and might be less sensitive to certain sentence structures or specific combinations of words which indicate a (call for) empathy.

Another reason why the OS models might perform better may be a sampling bias in the oversampling process. The oversampled training documents were created by taking the mean of a feature embedding of two documents in the minority class. The closest matching features were selected for each mean value, which might yield the same features as one of the seed documents. This could result in documents in the training dataset containing documents which are very similar to the original documents which

might be in the validation dataset. This would yield biased results in validation if the model is overfit on the training data, which many models were. If the models are indeed able to distinguish between original and oversampled texts, only the results of the models based on non-oversampled data should be regarded.

7.8 Relation to related work

In their sentiment classification study, Li et al. compare several BERT based models and LSTM models. The BERT model which was also used in this study - base BERT with a linear classification head - scored an F1 score of 0.60 and 0.73 on the two datasets they tested. The average F1 scores over the folds on the empathy detection task for both the NOS and OS models are consistently higher than this result, with F1 scores between 0.74 and 0.90. This is on average better than the best performing BERT model from Li et al., the BERT model with GRU head, which scored 0.61 and 0.75 on the two datasets. For the call for empathy classification task, the BERT model implemented in this study yields higher F1 scores than the BERT+GRU model implemented in Li et al.

Li et al. also compares LSTM models to the BERT based models and finds that the LSTM models perform overall worse than the BERT based models. Though the LSMT models compared in (Li et al., 2019) are different architecturally from the any of the LSTM models implemented in this study and therefore some difference is to be expected, the models from this study performed much worse than expected. The performance difference between the F1 scores of 0.55 and 0.66 obtained in (Li et al., 2019) and the score of 0.15 in this study is large.

In making these comparisons, it should be noted that the datasets used by (Li et al., 2019) are different in nature from the dataset used in this study, which is written by a different demographic about a much more personal subject. This makes the comparison of the model performance figures only partially valid.

Comparing the performance of the Khanpour et al.; Saeidi et al. models implemented in this study as well as the combined model with the performances of the LSTM models in the original papers also shows a large performance gap. The original Saeidi et al. model performed much better than any of the three LSTM models implemented in this study, with an F1 score of 0.69 on sentiment aspect classification. The Saeidi et al. as implemented in this study, though the best performing reference model, has a peak F1 score of 0.25 for the call for empathy classification task. The empathy classification task scores significantly worse with a peak of 0.15.

The ConvLSTM model in the contextually closer empathy classification from Khanpour et al. yielded an F1 score of 0.78. In addition, Li et al. also tested an LSTM model without convolutional layers, which performed well with an F1 score of 0.77. The Khanpour et al. ConvLSTM model as implemented in this study performed much worse than expected compared to the original results. The Khanpour et al. model did not train properly, as evidenced by the static loss figure and performance scores. The largest differences between the ConvLSTM model as implemented in this study and as implemented by Khanpour et al. were the dataset itself and the text embeddings. Whereas Khanpour et al. used pretrained Word2Vec embeddings, the embeddings used in this study were trained on the Kindertelefoon dataset itself. This smaller embedding training might lead to poorer embedding which could explain the overall poor performance of the LSTM models.

7.9 Limitations

In the assumptions that are made about the annotations, patterns in the data and the data itself, lie limitations and inaccuracies. In this section, some background behind the limitations which were not already covered in previous parts of the discussion is given.

One general limitation lies in the implementation of the proposed application of the language models for the Kindertelefoon. Prioritizing messages and identifying messages which are prone to abuse for volunteers working for the Kindertelefoon forum was realized primarily by classifying posts which call for an empathetic response through posts which express empathy. This approach has a fundamental flaw, in that it is dependant on other users to respond in an empathetic manner. Only when this is done, the algorithm can identify the post which calls for prioritization. To account for this, the direct prediction models were implemented. Since the direct prediction models perform better, these seem overall the better approach to the problem.

7.9.1 Data

In the modelling of the messages and in the annotation, each message was assumed to be either an expression of empathy or not. In reality, many messages contain more than merely an expression of empathy, a user might for example include an empathetic message for one aspect of a situation while simultaneously expressing that they cannot find themselves in another. To disambiguate this for the annotators, they were instructed to annotate every message which contains an expression of empathy as a message which is entirely an expression of empathy. In the models, this distinction is not so easily defined, as certain phrases in a message containing an empathetic expression might be indicative of a message which does not contain an expression of empathy. An architectural change in the models which could resolve this issue is to select the empathetic passage from each text. The question-answering head configuration of the BERT model is very suitable for this, but it would require a more extensive annotation scheme in which the empathetic phrases are selected from the texts. The model can then train to detect only the passages which call for or express empathy. This will likely increase the sensitivity for empathetic passages in longer responses which are not comprised completely of empathetic text.

One message label from the Kindertelefoon forum website which is not used as feature for the models is the 'best answer' label, as it was unused in large portions of the data. While the 'best answer' label may not be indicative of empathy directly, it may increase the confidence of a call for empathy classification based on the other features of a message marked as best answer. If a post labeled as 'best answer' is an empathetic response, the antecedent of this answer is more likely to be a post calling for an empathetic response than if a post that was not the best answer was used as reference. In other words, the 'best answer' posts can be used as weight in determining the class of the antecedent post.

Another feature from the forum that is not used in the models is the 'tag'. Tags can be created by the original poster of a thread to categorize it. Tags were not used as a feature because they were not consistently used and were not consistent across similar subjects and topics. Despite this, tags can give an insight into the topic of the thread, which is related to the probability of a message in that thread being an expression of empathy. For

this reason, the inclusion of tags might bring a performance increment to empathy classification. They can be implemented by reserving a set amount of dimensions in each feature vector which is optionally filled by the tags. Because of the varying length, there will often be some truncation or padding.

Barros et al. (2019) claim that ‘in the wild’ collected datasets suffer from a poor notion of what empathy is, as it is neither the target nor the empathizer who annotates the messages. They say this leads to an interpreted notion of empathy and unrealistic data. As the setup described is exactly the setup used in this study, the validity of the labels should be considered. The manual inspection of the annotators who scored poorly according to the annotation checking algorithm revealed that there was a large difference between the very poor annotators and fair annotators. The annotators which did not perform very poorly had a very broad definition of empathy, which included expressions of advice, though usually these included some element of cognitive empathy as well. In general, it is not feasible to manually check all annotations in a dataset of this size. The threefold annotation in combination with the annotator evaluation algorithm ensures that the majority of the three annotators for each message agree that it is an expression of empathy and that annotators have a high level of agreement. Nonetheless, the definition of empathy used by most of the annotators is broader than was defined beforehand.

7.9.2 Annotation

To reduce the complexity of the reply relations and to be able to infer a call for empathy from an empathetic response and a reply relation, each message is assumed to be a response to exactly one other post. In reality, the posts might be a reply to multiple posts or have no one particular antecedent. Some annotators asked how to deal with only being able to select one antecedent, and were instructed to find the closest match if no particular preceding post could be found or if the message in question could be a reply to multiple previous messages.

The empathy type label ranges from a cognitive awareness of another person’s feelings (low empathy) to experiencing feelings because of the other person’s feelings (high empathy). A lack of ‘is empathy’ label, indicating a complete lack of empathy in the message, can be seen as an extension of this, coming in before low empathy. This scale from no empathy to high empathy is not linear however, as ‘no empathy’ covers a very large range of emotional involvement. Both neutral and disdainful or contemptuous comments fall under the ‘no empathy’ label, while the scores do not reflect such a large difference in emotional involvement. To decrease the distance between different documents which fall under the ‘no empathy’ label and to decrease the potentially large difference between ‘no empathy’ and ‘low empathy’, a range of antonyms for empathy can be included in the labels, such as disdainful or contemptuous. The models can then also be trained to predict a value in this range instead of making a binary distinction. Furthermore, the classification of these properties in messages can be very useful on a forum, as these opposites of empathy often constitute messages which provoke a toxic culture on fora.

While it did not come up during the pilot test, some annotators indicated that there are a significant amount of messages in the Kindertelefoon data in which there is an empathetic response to messages which express fear. However, fear was not included as an empathy valence type, which caused an unknown amount

of messages to have an inaccuracy empathy valence label. Because being afraid of something is a likely root for a message on the Kindertelefoon forum and because empathetic responses are often appropriate to expressions of fear, this label should have been included.

One assumption that is made in the annotator comparison is that the proportions of good and poor annotators is consistent among the messages. Specifically, the method assumes that for every annotator, the proportion of co-annotators which are considered good annotators in relation to annotators which are considered bad annotators is similar. If this is not the case, and for example two bad annotators and one good annotator are paired up for a significant sequence of messages, all three will end up being classified as bad annotators. This is possible if all three annotators annotated in long consecutive stretches, which is how most of the annotators performed the task. Other conditions to this violation are that the annotators started roughly at the same message, which is down to chance, and that there were no other annotators annotating except for the three annotators which were paired up. This last condition is unlikely considering the temporal proximity of many of the annotations and the number of annotators who worked concurrently.

8 Conclusions

From the 169 annotators which annotated the Kindertelefoon data, 100 were found reliable through the iterative annotator reliability algorithm. The 69 dropped annotators annotated on average less than 10 messages each, so not many message annotations were dropped.

The reply relation algorithm was not able to reliably determine post antecedents and none of its specific components performed better than chance level. A direct prediction was preferred over the reply relation mediated model, as the latter performs at chance level because of the poor performance of the intermediary reply relation algorithm.

The BERT model was able to predict presence of empathy as well as call for empathy in the Kindertelefoon forum messages with high performance scores. The pretraining conditions for the BERT model did not effect this performance. Training the BERT model without updating transformer layers after pretraining yielded chance level results, indicating that the transformer layers encode task-specific patterns in the data.

The LSTM models were not able to classify either presence of empathy or call for empathy with a performance significantly better than chance level.

9 Summary

To classify messages which express empathy, an LSTM and a BERT based model were developed. To classify messages to which an empathetic response would be appropriate, the empathetic messages were classified and their antecedents were determined through a reply relation resolving algorithm. In addition to this, a direct prediction model was also developed for the call for empathy messages.

The reply relation components are in need of optimization. For most components, the inclusion criteria need to be improved. For the list type and the implied mention component, more patterns with which the relevant posts can be identified need to be provided. For the actual mention and similarity components, changes to the algorithm are proposed. For the question answer

component, a selection threshold needs to be optimized. In general, the basis of these components should be a useful starting point for producing a more effective reply relation algorithm.

The call for empathy classification based on the combination of reply relations and the empathy detection models does not perform well because of the poorly performing reply relations intermediary.

Ten epochs of masked language model (MLM) and sentence order prediction (SOP) tasks yielded eleven base BERT models, one for each pretraining epoch and one for no pretraining. The difference in task performance between the no pretraining condition and one epoch of pretraining was very high. The task performance on each subsequent epoch of pretraining was equal to the performance at epoch 1.

Because of the increased performance in the pretraining tasks, a better empathy classification performance was expected from models based on one epoch of pretraining than from models based on no pretraining. This difference did not appear, neither in the empathy classification task nor in the call for empathy classification task. To establish whether the trainable transformer layers were playing the same role as the pretraining in the model based on no pretraining, the models were trained again with frozen transformer layers, only training the classification head. The models trained with frozen transformer layers did not train on the task at all, which indicates that the transformer layers encode task related patterns and that the general natural language understanding which was supposed to be gained from the pretraining tasks did not transfer well to the classification tasks.

The BERT model architecture shows a high performance both in empathy classification and in call for empathy classification. The predictive power is high, and is reached after a small number of epochs, usually one epoch. Pretraining the BERT models with MLM and SOP tasks in addition to the Bertje pretraining from the base model did not improve performance significantly. Training only the classification head yields very poor results as the transformer layers in the BERT model encode patterns for the classification task, and not merely a generalizable natural language understanding.

The Saeidi et al. (2016) model and the combined LSTM model developed in this study performed poorly, and the Khanpour et al. (2017) model very poorly, on both the empathy classification and call for empathy classification tasks.

The Khanpour et al. (2017) and the Saeidi et al. (2016) models performed much worse on the Kindertelefoon data than on the datasets they were evaluated on in their original papers. The Saeidi et al. model did train but performed poorly nonetheless and the Khanpour et al. model did not train at all on either of the classification tasks.

10 Implications and future work

The annotation quality score which was generated for each annotator was used to distinguish between poor annotators and proper annotators in this study but this split does not need to be binary. The trustworthiness score can be applied as a weight in selecting the label from the threefold annotations. This weight ensures that the opinion of the most trustworthy annotator counts the most and gives a much more fine-grained input of annotator quality on the label selection. This selection prevents discarding annotators who have mediocre trustworthiness in order to gain high quality annotations, without a penalty in the annotation quality. This evaluation can be highly useful in selecting good

annotators from crowd-sourced datasets, but should be properly evaluated first. Simulated annotators with a distribution of annotator quality can be used to evaluate the performance of the annotator evaluation metric.

The models produced in this study are able to classify empathetic responses as well as messages which call for empathy. Besides alerting volunteers to messages which call for empathy, these classifiers can also be used to provide feedback to users who are drafting a response to a certain message. If this message is a call for empathy, the user might be prompted to provide an empathetic response, or consider their response in the perspective of the person they are replying to. The empathy classifier can provide feedback on how well the draft matches what can be considered an appropriate response.

A possible follow-up to this application is an active text prediction feature on the forum, which can provide users with a draft response based on previous messages in the thread, possibly in combination with previous messages of the responding user from similar threads. Architecturally, the LSTM model is more suitable for natural language generation than BERT is. However, since the performance of the BERT model was better than the LSTM model, a natural language generation model such as OpenAI GPT-3 which is more similar to BERT should be considered for this as well.

One of the motivations for producing a call for empathy classifier was the ability of Kindertelefoon volunteers to be able to prioritize some messages over others. While the call for an empathetic response does require a human response (versus for example a virtual conversational agent), there are more measures which should be taken into account. Problem severity and impact are important factors which should weigh into the prioritization as well, for which systems similar to the empathy and call for empathy classifier can be developed.

Not all features which were deemed important enough to be annotated were implemented in the model. As there were questions which arose from the simpler models in this project which caused more in-depth study of those models, such as the difference between the BERT model with and without trainable transformer layers, there was no room left in this project to incorporate the call to action, empathy type and empathy valence features. Implementing these features into the models opens up the ability to give a more fine-grained classification of messages instead of binary classifications. A further improvement can be made if the annotation scale is increased to incorporate items beyond ‘no empathy’ such as contemptuous or disdainful and if the missing forum-specific features such as tags and the ‘best answer’ label can be incorporated in a meaningful way.

Apart from being the mediator in the call for empathy classification, the reply relations could give insight into the message content by providing more detailed context. The message to which a given post is a response influences whether it could be considered empathetic. Likewise, a message could be considered toxic in one context and acceptable in another. To provide this context, the reply relation algorithm should be calibrated better.

11 Acknowledgements

I would like to thank Dr. Ing. Gwenn Englebienne and Dr. Hanane Ezzikouri for their supervision and guidance during this project. You gave shape to the study while letting me find my own way in planning and executing the work, which enabled me to make this work my own.

I am also very grateful to Dr. Mannes Poel, for his incredibly quick and accurate communication, grading and for providing me with access to very helpful resources. For contributing their expertise to this project and for their willingness to make time in doing so, I would like to thank Dr. Matthijs Noordzij and Prof. Dr. Gerben Westerhof.

References

- Alammar, J. (2018). The illustrated transformer. <https://jalammar.github.io/illustrated-transformer/>
- Aumayr, E., Chan, J., & Hayes, C. (2011). Reconstruction of threaded conversations in online discussion forums. *Fifth International AAAI Conference on Weblogs and Social Media*.
- Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Barros, P., Churamani, N., Lim, A., & Wermter, S. (2019). The omg-empathy dataset: Evaluating the impact of affective behavior in storytelling. *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, 1–7.
- Batson, C. D. (2009). These things called empathy: Eight related but distinct phenomena. In J. D. W. Ickes (Ed.), *Social neuroscience: the social neuroscience of empathy* (pp. 3–15). MIT press.
- Calvo, R. A., D'Mello, S., Gratch, J. M., & Kappas, A. (2015). *The oxford handbook of affective computing*. Oxford University Press, USA.
- Caplan, S. E., & Turner, J. S. (2007). Bringing theory to research on computer-mediated comforting communication. *Computers in human behavior*, 23(2), 985–998.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321–357.
- Chicco, D., & Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1), 6.
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Choi, J. D., Tetreault, J., & Stent, A. (2015). It depends: Dependency parser comparison using a web-based evaluation tool. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 387–396.
- Christov-Moore, L., Simpson, E. A., Coudé, G., Grigaityte, K., Iacoboni, M., & Ferrari, P. F. (2014). Empathy: Gender effects in brain and behavior. *Neuroscience & Biobehavioral Reviews*, 46, 604–627.
- Cong, G., Wang, L., Lin, C.-Y., Song, Y.-I., & Sun, Y. (2008). Finding question-answer pairs from online forums. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 467–474.
- Coulson, N. S. (2005). Receiving social support online: An analysis of a computer-mediated support group for individuals living with irritable bowel syndrome. *Cyberpsychology & behavior*, 8(6), 580–584.
- Cuff, B. M., Brown, S. J., Taylor, L., & Howat, D. J. (2016). Empathy: A review of the concept. *Emotion review*, 8(2), 144–153.
- Decety, J., & Jackson, P. L. (2004). The functional architecture of human empathy. *Behavioral and cognitive neuroscience reviews*, 3(2), 71–100.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- Eisenberg, N. (2000). Emotion, regulation, and moral development. *Annual review of psychology*, 51(1), 665–697.
- Eisenberg, N., Murphy, B. C., & Shepard, S. (1997). The development of empathic accuracy. *Empathic accuracy*, 73–116.
- El-Assady, M., Sevastjanova, R., Keim, D., & Collins, C. (2018). ThreadReconstructor: Modeling reply-chains to untangle conversational text through visual analytics. *Computer Graphics Forum*, 37(3), 351–365. <https://doi.org/10.1111/cgf.13425>
- Feng, S., Wang, Y., Liu, L., Wang, D., & Yu, G. (2019). Attention based hierarchical lstm network for context-aware microblog sentiment classification. *World Wide Web*, 22(1), 59–81.
- García-Pérez, R., Santos-Delgado, J.-M., & Buzón-García, O. (2016). Virtual empathy as digital competence in education 3.0. *International Journal of Educational Technology in Higher Education*, 13(1). <https://doi.org/10.1186/s41239-016-0029-7>
- Goleman, G. (1995). *Emotional intelligence – why it can matter more than iq*. Bantam Books, New York.
- Håkansson, J., & Montgomery, H. (2003). Empathy as an interpersonal phenomenon. *Journal of Social and Personal Relationships*, 20(3), 267–284. <https://doi.org/10.1177/0265407503020003001>
- Hart, D., & Fegley, S. (1995). Prosocial behavior and caring in adolescence: Relations to self-understanding and social judgment. *Child development*, 66(5), 1346–1359.
- Haugen, P. T., Welsh, D. P., & McNulty, J. K. (2008). Empathic accuracy and adolescent romantic relationships. *Journal of adolescence*, 31(6), 709–727.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.
- Iacoboni, M. (2005). Understanding others: Imitation, language, empathy. *Perspectives on imitation: From cognitive neuroscience to social science*, 1, 77–99.
- Kalliopuska, M. (1983). Empathy in school students.
- Kassin, S., Fein, S., Markus, H. R., McBain, K. A., & Williams, L. (2019). *Social psychology international edition*. Cengage.
- Khanpour, H., Caragea, C., & Biyani, P. (2017). Identifying empathetic messages in online health communities. *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 246–251.
- Kim, S. N., Wang, L., & Baldwin, T. (2010). Tagging and linking web forum posts. *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, 192–202.
- Kindertelefoon. (2015). Gebruikersnaam aanpassen. <https://forum.kindertelefoon.nl/over-de-kindertelefoon-54/gebruikersnaam-aanpassen-14651>
- Kindertelefoon. (2017). Profiel thread. <https://forum.kindertelefoon.nl/over-de-kindertelefoon-54/profiel-25403>
- Kindertelefoon. (2018). Veelgestelde vragen - hoe werkt het forum? <https://forum.kindertelefoon.nl/over-de-kindertelefoon-54/veelgestelde-vragen-hoe-werkt-het-forum-36130>
- Levenson, R. W., & Ruef, A. M. (1992). Empathy: A physiological substrate. *Journal of personality and social psychology*, 63(2), 234.
- Li, X., Bing, L., Zhang, W., & Lam, W. (2019). Exploiting bert for end-to-end aspect-based sentiment analysis. *arXiv preprint arXiv:1910.00883*.
- Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- Madabushi, H. T., Kochkina, E., & Castelle, M. (2020). Cost-sensitive bert for generalisable sentence classification with imbalanced data. *arXiv preprint arXiv:2003.11563*.
- Niedenthal, P. M., Mermillod, M., Maringer, M., & Hess, U. (2010). The simulation of smiles (sims) model: Embodied simulation and the meaning of facial expression. *Behavioral and brain sciences*, 33(6), 417.
- Olah, C. (2015). *Understanding lstm networks*. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Oostdijk, N., Reynaert, M., Hoste, V., & Schuurman, I. (2013). The construction of a 500-million-word reference corpus of contemporary written dutch. *Essential speech and language technology for dutch* (pp. 219–247). Springer, Berlin, Heidelberg.
- Opentaal, S. (2020). <https://www.opentaal.org/>
- Ordeman, R., de Jong, F., Van Hessen, A., & Hondorp, H. (2007). Twnc: A multifaceted dutch news corpus. *ELRA Newsletter*, 12(3/4), 4–7.
- Peer, E., Brandimarte, L., Samat, S., & Acquisti, A. (2017). Beyond the turk: Alternative platforms for crowdsourcing behavioral research. *Journal of Experimental Social Psychology*, 70, 153–163.
- Pfeil, U., & Zaphiris, P. (2007). Patterns of empathy in online communication. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 919–928.
- Preece, J. (1999). Empathy online. *Virtual Reality*, 4(1), 74–84. <https://doi.org/10.1007/bf01434996>
- Preece, J. J., & Ghozati, K. (2001). Experiencing empathy online. *The internet and health communication: Experiences and expectations* (pp. 237–260). SAGE Publications, Inc. <https://doi.org/10.4135/9781452233277.n11>
- Renda, A. (2018). The legal framework to address “fake news”: Possible policy actions at the eu level. [https://www.europarl.europa.eu/RegData/etudes/IDAN/2018/619013/IPOL_IDA\(2018\)619013.EN.pdf](https://www.europarl.europa.eu/RegData/etudes/IDAN/2018/619013/IPOL_IDA(2018)619013.EN.pdf)
- Saeidi, M., Bouchard, G., Liakata, M., & Riedel, S. (2016). Sentihood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods. *arXiv preprint arXiv:1610.03771*.
- Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
- Shrestha, L., & McKeown, K. (2004). Detection of question-answer pairs in email conversations. *Proc. of COLING*.
- Silke, C., Brady, B., Boylan, C., & Dolan, P. (2018). Factors influencing the development of empathy and pro-social behaviour among adolescents: A systematic review. *Children and Youth Services Review*, 94, 421–436.
- Spencer, R., Pryce, J., Barry, J., Walsh, J., & Basualdo-Delmonico, A. (2020). Deconstructing empathy: A qualitative examination of mentor perspective-taking and adaptability in youth mentoring relationships. *Children and Youth Services Review*, 105043.
- Spring, T., Casas, J., Daher, K., Mugellini, E., & Abou Khaled, O. (2019). Empathic response generation in chatbots. *SwissText*.
- Stern, J. A., & Cassidy, J. (2018). Empathy from infancy to adolescence: An attachment perspective on the development of individual differences. *Developmental Review*, 47, 1–22.

Classification of a Call For Empathy in Child Help Forum Messages.

- Strauss, C., Taylor, B. L., Gu, J., Kuyken, W., Baer, R., Jones, F., & Cavanagh, K. (2016). What is compassion and how can we measure it? a review of definitions and measures. *Clinical psychology review*, 47, 15–27.
- Sun, C., Huang, L., & Qiu, X. (2019). Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. *arXiv preprint arXiv:1903.09588*.
- Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 3104–3112.
- Tulkens, S., Emmery, C., & Daelemans, W. (2016). Evaluating unsupervised dutch word embeddings as a linguistic resource. In N. C. (Chair), K. Choukri, T. Declerck, M. Grobelnik, B. Maegaard, J. Mariani, A. Moreno, J. Odiijk, & S. Piperidis (Eds.), *Proceedings of the tenth international conference on language resources and evaluation (lrec 2016)*. European Language Resources Association (ELRA).
- Underwood, B., & Moore, B. (1982). Perspective-taking and altruism. *Psychological bulletin*, 91(1), 143.
- Van Tilburg, M. A., Unterberg, M. L., & Vingerhoets, A. J. (2002). Crying during adolescence: The role of gender, menarche, and empathy. *British Journal of Developmental Psychology*, 20(1), 77–87.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 5998–6008.
- Vries, W. d., Cranenburgh, A. v., Bisazza, A., Caselli, T., Noord, G. v., & Nissim, M. (2019). BERTje: A Dutch BERT Model. *arXiv:1912.09582 [cs]*. <http://arxiv.org/abs/1912.09582>
- Wang, H., Wang, C., Zhai, C., & Han, J. (2011). Learning online discussion structures by conditional random fields. *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 435–444. <https://doi.org/10.1145/2009916.2009976>
- Wang, Y., Joshi, M., Cohen, W. W., & Rosé, C. P. (2008). Recovering implicit thread structure in newsgroup style conversations. *ICWSM*.
- Wolf, F., & Gibson, E. (2005). Representing discourse coherence: A corpus-based study. *Computational linguistics*, 31(2), 249–287.
- Xi, W., Lind, J., & Brill, E. (2004). Learning effective ranking functions for newsgroup search. *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 394–401.
- Xiao, B., Imel, Z. E., Georgiou, P. G., Atkins, D. C., & Narayanan, S. S. (2015). "rate my therapist": Automated detection of empathy in drug and alcohol counseling via speech and language processing. *PloS one*, 10(12).
- Yang, K., Lee, D., Whang, T., Lee, S., & Lim, H. (2019). Emotionx-ku: Bert-max based contextual emotion classifier. *arXiv preprint arXiv:1906.11565*.

Appendices

A Interview setup

To establish a well-grounded definition of empathy which is appropriate for the target age bracket for the Kindertelefoon, two interviews and a focus group are planned in which the topic is discussed. The interviews are conducted with a health psychologist and a child-development psychologist to get an insight into the psychological perspective on empathy as well as the developmental stage the children who visit the Kindertelefoon forum are in. The focus group is conducted with a small number of Kindertelefoon volunteers, which gives provides the perspective on empathy of those who stand closest to the subject matter.

The goals of the interviews and the focus group are to get a psychological and Kindertelefoon perspective of

- A definition of empathy
 - Along with (groups of) constituents
 - Along with pointers and suggestions for recognizing empathy and calls for empathy
 - Along with suggestions for examples for (groups of) constituents
- Nuances in the definition of empathy
 - In the age bracket in this study (12-18) this is important to differentiate from literature definitions which often concern adults
 - In these matters for online expressions of empathy

Specifically for the Kindertelefoon, these sessions aim to investigate

- What the differences between the different kindertelefoon media (both from volunteer and help-seeker perspective) are
- How priorities are selected by volunteers, and how volunteers select which post to comment on or not
- How chatbots can contribute to the environment

and to validate assumptions made:

- About post types
- About dependency structures for different post types
- About dependency structures in general

A.1 General questions

Tijdens het voorstellen en de uitleg over het project worden ten minste de volgende onderwerpen behandeld:

- De bedoeling is dat er uiteindelijk een classificatie gebouwd wordt die kan ondersteunen in het vinden van posts waarop met prioriteit gereageerd moet worden door posts te herkennen waar een empathische reactie op zijn plaats is
- Het doel is niet om medewerkers te vervangen maar om ze te ondersteunen door aan te geven welke posts de aandacht van een mens nodig hebben
 - Mening over dit doel?

Om het concept van empathie in het algemeen te definiëren:

- Welke constructen liggen ten grondslag aan empathie?
 - Niet alleen cognitieve/affectief onderscheid maar ook:
 - Bijvoorbeeld een sociale staat delen, mimiek nabootsing, fysiologische reactie, perspectief nemen
 - Zie de acht voorbeelden

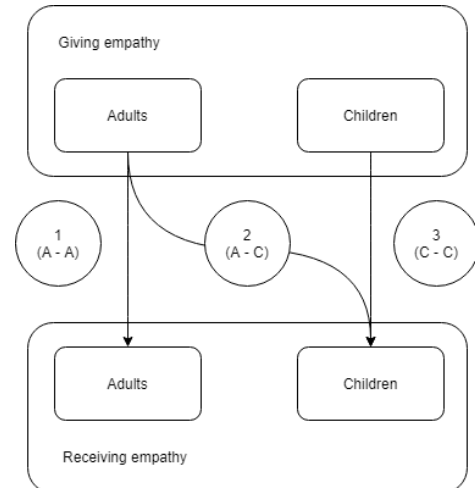


Figure 48: Graph indicating different empathizer-target relationships. Constructs of empathy can be places along these lines to indicate which constructs apply to which relationship.

- Card sorting: groepeer de acht empathie constructen uit sectie 2.1 in zo weinig mogelijk groepen zonder een onderscheidende factor van de groep te verliezen

Hoe is empathie te herkennen?

- Is empathie tekstueel vast te stellen of is het beter op te merken uit intonatie, intentie en context?
- Hoe zou je te werk gaan bij het herkennen van een empathische reactie vs. een niet-empathische reactie?
- Zit er verschil in empathische uitingen van volwassenen tegenover kinderen versus tegenover andere volwassenen?
- En zit er een verschil in empathische uitingen van kinderen (12-18) tegenover andere kinderen versus volwassenen tegenover kinderen.
 - Eerder opgeschreven constructen in empathy interview diagram plaatsen, *geen* rekening houdend met een online context
 - Eerder opgeschreven constructen in empathy interview diagram plaatsen, *wel* rekening houdend met een online context

Hoe is een vraag naar empathie te herkennen?

- Wanneer is een empathische reactie op zijn plek? Bijvoorbeeld in de context van het kindertelefoon forum
Toelichting: verschillende soort posts op Kindertelefoon forum
- Specifiek: hoe zou een kind (12-18) een vraag naar empathie formuleren? Bijvoorbeeld door een verhaal te verzinnen (asking for a friend)

A.2 Focus group Kindertelefoon

Naast de vragen uit het algemene deel worden er ook specifieke vragen aan de Kindertelefoon medewerkers gesteld tijdens de focus groep.

Algemeen over Kindertelefoon:

- Hoe ziet een dag voor een medewerker er uit?
 - Ben je vooral bezig met chat/bellen/forum?
- Wat zijn verschillen tussen chat, telefoon en het forum?

Classification of a Call For Empathy in Child Help Forum Messages.

- Zijn er veel kinderen die eerst op het forum wat posten en dan toch bellen of andersom?
- Hoe worden de verschillende media ervaren?
 - Door de kinderen
 - Door de medewerkers
- Wat gebeurt er als er geen medewerkers beschikbaar zijn?

Post prioriteit

- Hoe bepaal je welke posts je op reageert en welke niet?
- Bekijk je (bijna) alle posts?

Chatbot

- Zou een chatbot een positieve bijdrage kunnen zijn als er geen medewerkers beschikbaar zijn voor de chat?
- En als er wel medewerkers beschikbaar zijn? (dus in het algemeen)

Empathie

- Wat betekent empathie voor jou/jullie?
- Wat zijn belangrijke verschillen tussen reacties van andere kinderen en medewerkers op het kindertelefoon forum?
- Welke verschillen zien jullie tussen reacties van kinderen op het forum?

Validatie:

- Wat vind je van de indeling van post types: vragenlijst, specifieke hulpvraag en emotionele ontlading?
- Zie je vaak empathische reacties bij vragenlijst type posts?
- Hoe zit de post reactie structuur van de verschillende post types in elkaar

A.3 Interview psychologen

Wat zijn handvatten die aangeboden kunnen worden aan kinderen tussen 12 en 18 met problemen thuis, onzekerheden en vragen die ze niet aan een persoon in hun real life omgeving kunnen vragen? (hoe) kan empathie bijdragen aan de emotionele ontwikkeling als er problemen zijn in thuissituatie?

B Empathy components overview

Empathy as knowing another person's internal state

The first definition of empathy is a cognitive one and as such is also known as 'cognitive empathy' or 'empathetic accuracy'. Defining empathy as knowing another person's internal state refers to being aware, through linguistic or nonverbal communication, of what is on the other person's mind. This notion is the first of Håkansson and Montgomery's constituents of empathy. It may not be accurate or complete, this definition merely requires an active awareness of one person's belief of another person's internal state.

Empathy as physical mimicry

A more neurological perspective of empathy is based in physical mimicry. This view denotes that empathy is gained from purposeful simulation of another person's (facial) expression or that empathy necessarily coincides with neurological and physical mimicry Niedenthal et al., 2010. The core concept in this perspective is that the embodiment of an emotion causes neurological pathways to activate similarly to the way they would if the person was the primary experiencer of the emotion. This gives an impression of what another person is feeling through the vicarious experience.

Empathy as feeling how another person feels

An affective perspective of empathy is coming to feel as another person is feeling. This is more than merely knowing another person's internal state and requires more than merely physical mimicry. This definition is based on *experiencing* the emotion that another person is having. This concept of feeling how another person feels is usually known as empathetic contagion Calvo et al., 2015 or outside psychology as sympathy Batson, 2009.

Empathy as imagining how another is thinking and feeling

Although seemingly similar to the first (cognitive) definition, imagining how another is thinking and feeling extends merely concluding how the other feels with imagination based on what is known from previous experiences with that person or with other people. This is not necessarily based on one's own experiences or character but rather on what the perspective taker thinks the other person experiences.

Empathy as literally perspectivising

A somewhat archaic but still well-known perspective is literal perspective taking. Here, one tries to not only take perspective in the situation of another person but also to reason the way that person would reason. This involves an extensive perspective taking ability to the point in which it is unreasonable to assume this approach might be actually feasible. Rather, the core principal is to get as many contextual factors correct in empathising.

Empathy as imagining how one would feel in another person's place

A view which is often referred to by the term 'perspective taking' is to imagine how one would behave and feel in another person's place. This is different from imagining how another is thinking or feeling and from literally perspectivising as imagining in

place is based on one's own experiences and character in another person's situation instead of the other person's character. This is the third constituent of Håkansson and Montgomery's study Håkansson and Montgomery, 2003. The active reflection on one's past experiences contributes to the connectedness with another person, as commonalities are sought which may shed light on how one would act or feel in another's place Spencer et al., 2020.

Empathy as feeling distress because of another person's malaise

Distinct from feeling distress *with* another person because of perspective taking, empathy as feeling distress as a result of witnessing another person's suffering has also been used as a definition of empathy. This concept is also known as 'empathetic distress'.

Empathy as feeling for another person's suffering

A perspective based in a more altruistic sense than the other definitions, empathy is also defined as feeling distress or discomfort because of another person's distress. This perspective is different from *feeling how another person feels*, as the reactionary emotion does not need to be the same. This is the forth constituent of empathy according to Håkansson and Montgomery.

C Agreement scores full figures

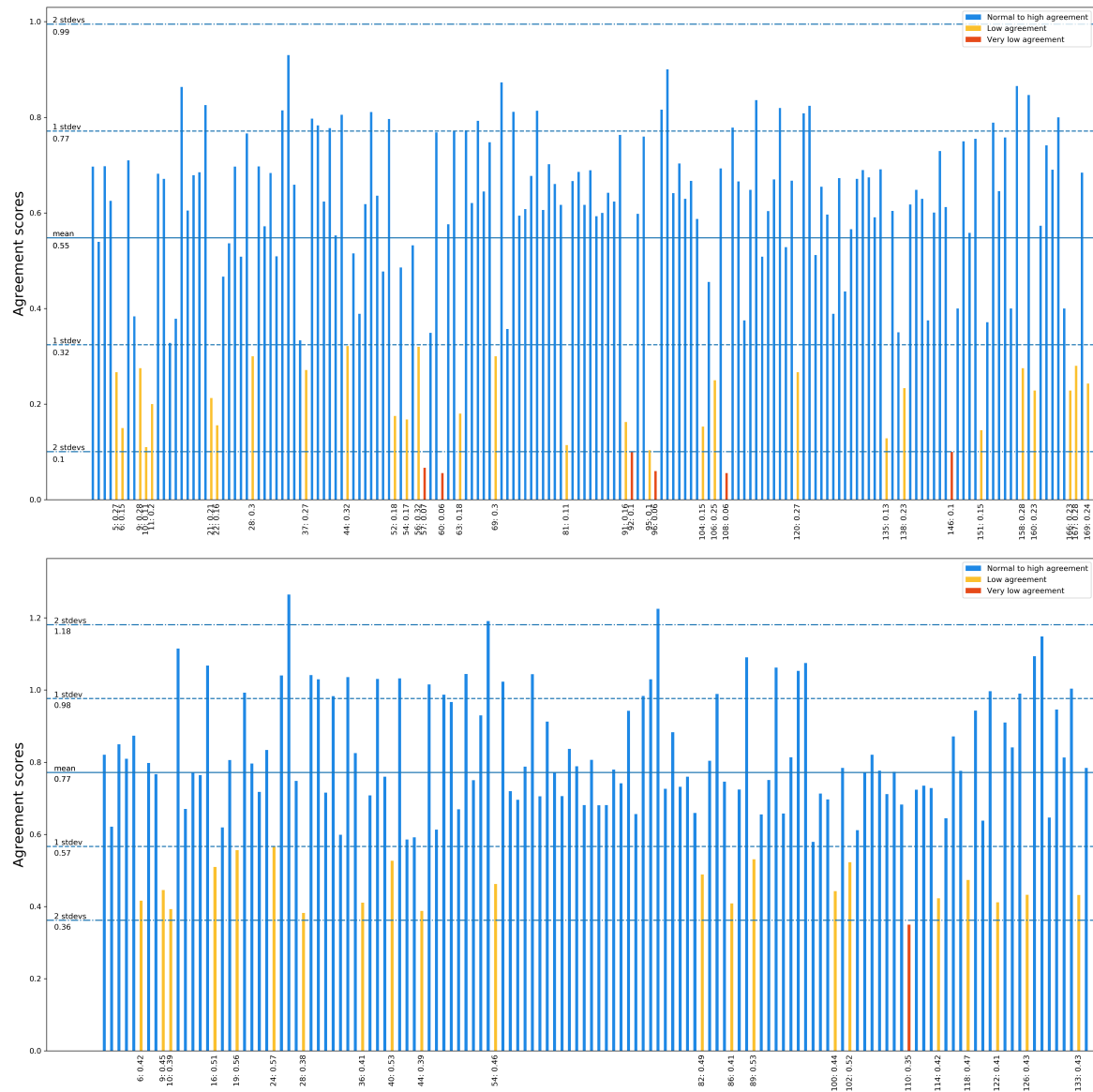


Figure 49: Agreement scores for iterations 1 and 2

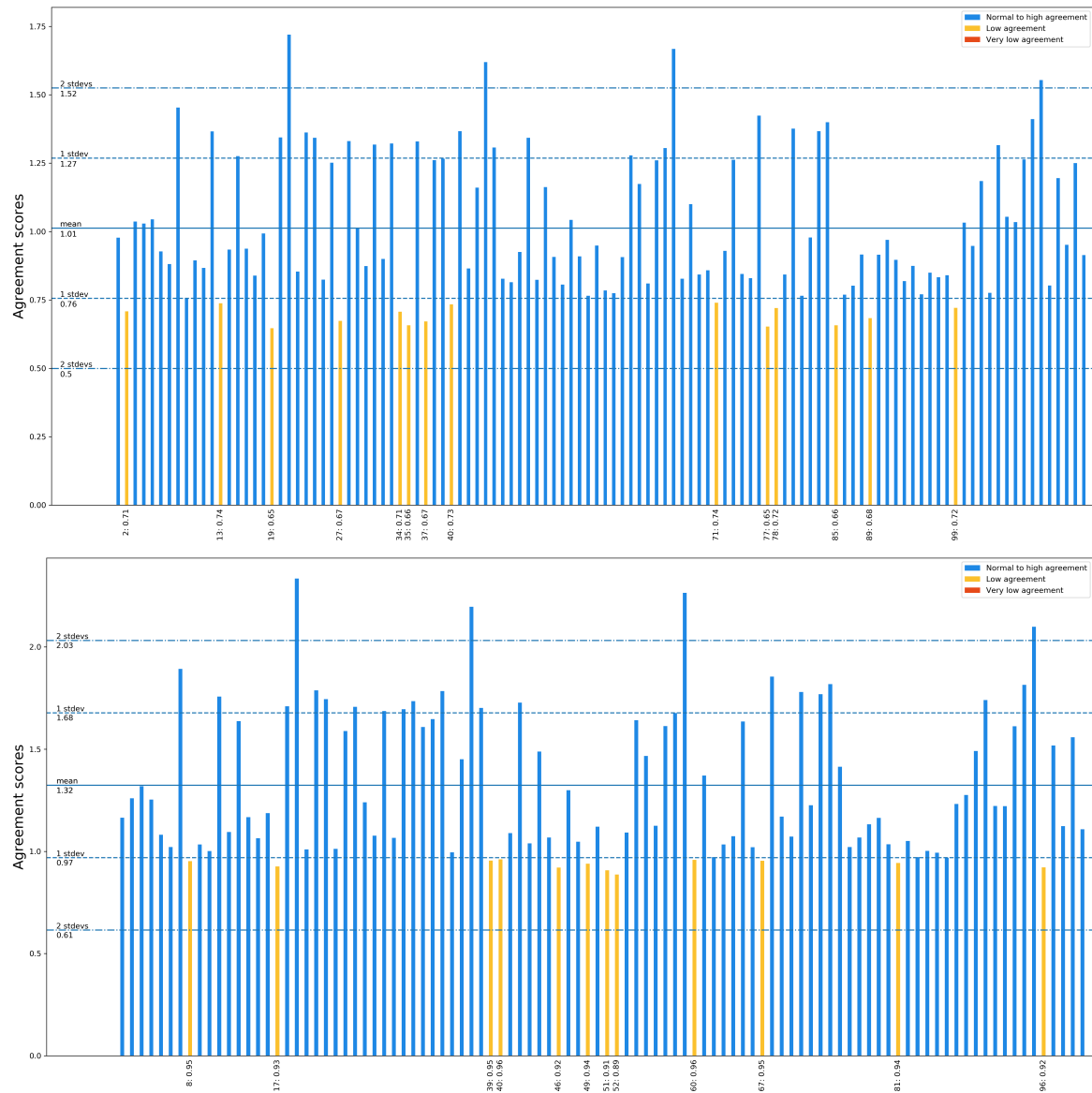


Figure 50: Agreement scores for iterations 3 and 4

D Annotation applications

The first iteration of the annotation program was a terminal based application. It iterated through posts within a thread and prompted the annotator with questions about the post. For each post, the annotator was asked whether that post expressed empathy or not, whether it called for an empathetic response or not, and, for all but the first two posts in each thread, to which post the current post was a response. At the end of every thread, the annotator was asked if they would like to continue or stop after that thread. Figure 51 shows a screenshot of the terminal application.

```
Hij is met 16 zegt hij en ik ben 13

post number 5
Ja ik heb gevraagd of hij zich niet als iemand anders voor doet maar sinds die
skipt hij me

post number 6
Hoi Femke09312! Je moet hiermee naar de politie gaan, en vertel het je ouders.
Op die manier bescherm je jezelf, en ook andere meisjes die hier in de
toekomst nog in aanraking komen. Hij is schip, is het eigenlijk een bewijs dat
hij het is. Want dat is een teken dat hij bang is dat je het gaat laten weten
aan de politie. Heb je hem je adres gegeven of gezegd in welke stad je woont?
Want als je dat niet hebt gedaan is het niet veilig en kan hij je niet terug
vinden, wat je weleens bij andere verhalen hoort. Heb je dat wel gedaan hoe
je niet meteen bang te worden. Want als je het dan vlug aan de politie meld en
erbij zegt dat hij je stadi adres weet, gaan ze beter op je letten. Ik hoop
dat ik je een beetje kan helpen! En laat me gerust weten als ik je met iets
anders misschien kan helpen! Xx sunflower!

post number 7
bobbbaas schreef: hoi als eerste raad ik je aan om deze persoon te vragen of
dat zijn echte naam is en of hij dat inderdaad is. daarna raad ik je aan om er
mee naar de politie te gaan (dan anoniem als het goed is) het is namelijk
illegaal als hij het echt is ik snap dat je dit niet zou willen maar het is
het beste wat je kan doen op dit moment. Ja ga alsjeblieft naar de politie, je
weet je zelf het, maar moet je na gaan, hij doet dit niet alleen bij jou. Dit
soort viespeuken moeten van de normale populatie wegblijven. Ga naar de
politie en die zorgen er wel voor dat hij naar de gevangenis moet of een hele
erger straf krijgt. Alsjeblieft ga naar de politie!


Currently annotating post #8

Hi, als hij die vraag van je ontwijkt is het al wel duidelijk dat hij
waarschijnlijk niet is wie hij zegt dat hij is! het is illegaal als hij deze
foto's/filmpjes heeft en al helemaal als hij is. Probeer meer van deze
persoon te weten te komen (waar komt hij vandaan/e-mail/telefoonnummer) dit
kan allemaal belangrijk zijn in een politieonderzoek. Geef natuurlijk niet jou
nummer, adres en e-mail. Probeer het ook aan je ouders te vertellen en dat je
misschien aangifte wilt doen... hoe moeilijk het ook is ze zullen je steunen in
zo'n situatie. Hoop dat je er wat mee kan! Stuur me een berichtje als je me
nodig hebt of vragen hebt x Angel

? Does this post express empathy? No
? Does this post ask for empathy? No
? Type in the number of the post this post replies to. 1
? You have finished thread help-mij-hiermee-plis-48887, do you want to continue? (Y/N)
```

Figure 51: First version of annotation tool, a terminal application

The final version of the annotation tool is a website on which anybody who is willing to annotate can view messages and provide information about the posts. The landing page functioned as an information brochure and a place to indicate consent to take part in the annotation. Figures 52 and 53 show screenshots of the website.


Theng

Hallo!

Ontzettend bedankt voor het helpen bij mijn afstudeerproject. Het doel van deze studie is om een programma te ontwikkelen dat berichten die geplaatst zijn op het forum van de Kinderhelpline kan interpreteren. Om precies te zijn is het doel een bericht te voorzien van een empathische reactie op basis van de inhoud van het bericht. Dit kan gebruikt worden om medewerkers van de Kinderhelpline een beter overzicht te geven welke berichten met aandacht behandeld moeten worden en eventueel om een voorstel voor een bericht te genereren. Om dit te kunnen bereiken heb ik informatie nodig van posts op het forum uit het verleden, hierbij kan je me helpen.

Op deze website krijg je forumposts van het Kinderhelforum te zien. Over deze posts worden een aantal vragen gesteld. Deze vragen zijn te beantwoorden help je me enorm met het bouwen van mijn programma. Alleen als je vragen een (gebruikers)naam in te vullen. Dit heeft niet je echte naam te zijn als je een naam invult. Het invullen van een naam helpt me verschillende antwoorden uit elkaar te houden en zorgt er voor dat je reactie kan gaan naar je gebruikersnaam als je de site sluit.

Bij het antwoorden van de berichten zal elke keer een bericht ingevuld worden. Dit bericht is aangegeven met een groene pijl. Onder het bericht staan enkele veldjes, deze veldjes zijn de gegevens die het bericht van toepassing zijn. Deze veldjes zijn er zo uit:

☐ Dit bericht is een uitdrukking van empathie.

Als een bericht een uitdrukking van empathie is, wordt er gevraagd om het soort empathie en de correcte emotie bij het bericht aan te geven. Dit doet er zo uit:

Soort empathie in het bericht:

☐ Ik begrijp dat je zo voelt

☐ Ik wil me zelf ook zo

☐ Ik wil me zo door de situatie die jij hebt meegemaakt

Emotie bij het bericht:

☐ blij

☐ geschokt

☐ boos

Je ziet vervolgens een voorbeeld van de gegevens die je moet invullen. Als je bijvoorbeeld optie 1 bij het soort empathie hebt aangevinkt en optie 1 bij de emotie doet dat er zo uit:

Het bericht ligt op: Ik begrijp dat je zo voelt

Als er meerdere reacties geplaatst zijn in een bepaalde thread wordt er gevraagd om aan te geven welk eerste bericht het huidige bericht op reageert. Dit kan door onderaan de pagina het berichtnummer in te vullen of op het eerste bericht te klikken. Dit kan ook op de eerste post worden gedaan. Het bericht dat je een bericht hebt gereageerd op.

Je kunt zelf bepalen hoe lang je me helpt. Als je tussentijd wil stoppen en later verder wil gaan, gebruik dan de laatste berichtnummers. Zo voorkom je dat je mogelijk dezelfde berichten te zien krijgt.

Belangrijk: Deze berichten zijn niet alleen door mij gebruikt, het kan zijn dat er wetgeving of andere wetten van toepassing zijn op de berichten. Als dit het geval is kan je contact met me opnemen.

Mocht je vragen hebben over het verloop of de resultaten van het onderzoek of je de naam wilt wijzigen of de gegevens wilt verwijderen, neem dan contact met mij op via mijn email adres. Ik heb het niet gewend te vragen of opmerkingen over het verloop van het onderzoek te richten aan de onderzoeksleider kan contact gezocht worden met Geert Engelen.

NB: Om deze site te laten werken wordt er een cookie opgeslagen. Bij het invullen van een naam ga je akkoord met het plaatsen van een cookie.

Heel erg bedankt voor je hulp!

☐ Ik ga akkoord met het plaatsen van een cookie en met de deelname aan het onderzoek.

Naam/SONA nummer/Participant ID:

Opslaan en doorgaan




Theng


Figure 52: Landing page of the website, including information brochure, username input and informed consent.

UNIVERSITY OF TWENTE.


Gebruikersnaam: Qwerty
Thread naam: 5-keer-met-de-dood-bedreigt-politie-of-niet-10367

Vorige berichten in thread:

 Bericht nummer 1 geschreven door een verwijderde gebruiker:
Hey allemaal, Laatste ik lig nu elke dag met de dood bedreigt en gestakt, nu is het zo dat ik weet wie het is, moet ik naar de politie gaan of moet ik het zelf oplossen? Grout, Doudveld

 Bericht nummer 2 geschreven door een verwijderde gebruiker:
Vat is het dan voor emmer? Een klagenood? Een en? Je zegt "hadt", bedoel je dat het nu niet meer gebeurt? Dan is het al opgelost toch? En wat voor oplossing bedoel je met "zelf oplossen"?

Post nu aan het annoteren, geschreven door een verwijderde gebruiker:

 Als je met de dood wordt bedreigt moet je aangifte doen bij de politie. De politie neemt doodbedreigingen namelijk altijd serieus! Je kunt dit niet zelf oplossen.

☒ Dit bericht is een uitdrukking van empathie.

Gebruik de empathie opties en emote opties om de kern van het bericht samen te vatten. Onder de opties staat een voorbeeld een bericht met de gekozen opties.

Soort empathie in het bericht

☐ Ik begrijp dat je je zo voelt

☐ Ik kan me voorstellen dat je je zo voelt

☐ Ik zou me zo voelen als ik in jouw schenen stond

☐ Ik voel me zelf ook zo

☐ Ik voel me zo door de situatie waarin je je bevindt

☐ Ik voel me zo door wat je hebt meegemaakt

Emote bij het bericht

☐ blij

☐ opgevoelt

☐ emotioneel

☐ geraakt

☐ gekust

☐ verdrietig

☐ boos

Het bericht ligt op:

☐ Een empathische reactie is op zijn plaats als reactie op dit bericht.

☐ Er wordt in dit bericht een vraag gesteld.

☐ Er wordt in dit bericht een vraag beantwoord.

☐ Er wordt in dit bericht een actie aanbevolen.

Je hebt bericht nummer 2 geselecteerd.

2

Als het bericht een uitdrukking van empathie is moeten de soort en de emote aangegeven worden.


 Ting

Figure 53: Example of a post annotation, with reply highlighted in green.

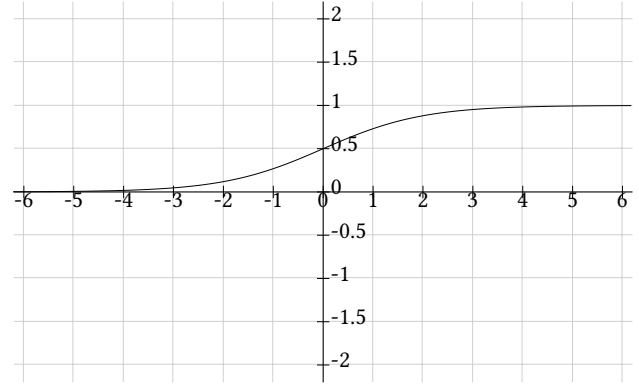
E Activation functions

Activation functions remap ranges of values to other ranges. Traditionally, nonlinear functions such as the sigmoid function and the hyperbolic tangent function (see equation 13 through 15) are used because they allow the representation of more complex functions. The sigmoid function (represented in this report as σ) maps input values to a range between 0 and 1, whereas the hyperbolic tangent function (\tanh) maps values between -1 and 1. One downside to these functions is that values larger than 2 or smaller than -2 (for \tanh) are mapped to a very small portion of the distribution. This may cause the activation functions to become saturated, where all values are on end of the distribution. To counter this, the rectified linear unit is used (see equation 15). This activation function simply returns the input values if it is positive, else it returns 0.

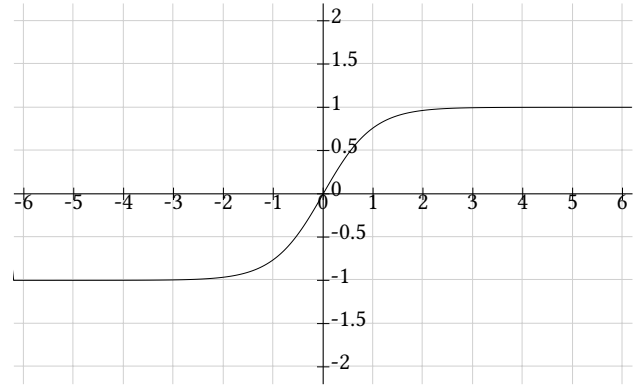
$$\sigma = \frac{e^x}{e^x + 1} \quad (13)$$

$$\tanh = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (14)$$

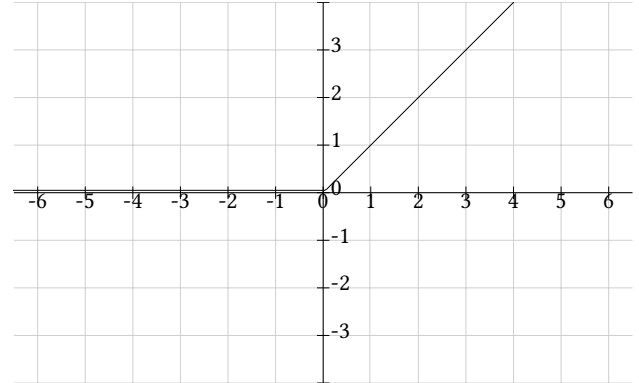
$$\text{relu} = \max(0, x) \quad (15)$$



(a) Sigmoid distribution.



(b) Hyperbolic tangent distribution.



(c) ReLU distribution.

Figure 54: Distributions of the three activation functions used in this study.

F Top 30 antecedent label and reply resolution prediction counts

	Antecedent label	reply resolution prediction
0	3050	3482
1	1130	1130
2	560	528
3	445	386
4	295	260
5	92	214
6	183	129
7	139	108
8	108	77
9	61	35
10	66	39
11	53	43
12	47	39
13	44	17
14	39	32
15	35	19
16	27	19
17	10	14
18	22	16
19	21	25
20	15	12
21	20	17
22	14	15
23	16	17
24	9	7
25	16	9
26	16	17
27	6	5
28	9	12
29	8	17
30	10	4

To improve automated detection of empathetic expressions and to streamline online discussion board moderation, an LSTM and a BERT neural network were trained to detect empathetic responses and calls for an empathetic response. Messages from the Kindertelefoon forum, labeled using crowd sourcing, were used as case study to provide a proof of concept. Assessing annotator reliability and determining reply relations were core considerations in cleaning the data. The BERT and LSTM models were trained on empathy detection and on call for empathy detection directly. The BERT model performed well in both the empathy and call for empathy classification tasks, outperforming the LSTM models. The empathy classification and direct call for empathy classification models using BERT constitute a new state of the art in text-based empathy modelling and text-based emotion classification systems in general.

