Institute for Natural Language Processing
University Stuttgart
Pfaffenwaldring 5B
D-70569 Stuttgart

Multitask Classification across Psychological Models of Emotion and Affect

Hai Dang Nguyen

Bachelor Thesis

Examiner:	Prof. Dr. Sebastian Padó
Supervisor & Examiner:	Dr. Roman Klinger

Start date:	10.03.2020
End date:	10.09.2020

Abstract

The classification of emotions from text is usually done on basis of a single psychological model of emotion and affect with a singletask classifier. The goal of this bachelor thesis is to evaluate the merit of multitask learning across emotion models. Here, a multitask approach is taken, where multiple data sets, each annotated on basis of a different psychological model, are trained in a multiheaded neural network. Comparing the F1-score / mean-squared-error of the multitask classifiers to the singletask classifiers show no improvement.

Abstract

Die Klassifizierung von Emotionen aus Text wird normalerweise auf Basis eines einzelnen psychologischen Modells von Emotion und Affekt mit einem singletask Klassifizierer erwirkt. Das Ziel dieser Bachelorarbeit ist es, den Vorzug vom multitask Lernen von Emotionsmodellen zu evaluieren. Hierbei wird ein multitask Ansatz genommen, bei dem mehrere Datensätze, die jeweils auf Basis unterschiedlicher psychologischer Modelle annotiert wurden, in einem mehrköpfigen neuronalem Netzwerk trainiert werden. Der Vergleich des F-Maß / mittleren quadratischen Fehlers der multitask Klassifizierer und der singletask Klassifizierer zeigt keine Verbesserung.

Contents

1 Introduction

Emotion analysis is the identification of emotions, it answers the question of what feelings lie behind a certain action, picture or other media. Research on emotions is being done in multiple fields including computer science. One research topic is the extraction of emotion from text. Here, scientists build and train a classifier to match an emotion to a text input. That way, they try to answer various research questions, like how well specific learning approaches perform or how different emotions are expressed on different corpora (Roberts et al. (2012)).

Compared to sentiment analysis, the results of emotion analysis is more nuanced. While sentiment analysis can find out what opinion or sentiment is stated (either positive or negative), emotion analyses the underlying emotions. As an example, sentiment analysis on user reviews can detect that users are reporting negative reviews, while emotion analysis can determine whether the users are frustrated, disappointed or angry, which makes it a useful tool for companies to understand the costumer, according to Yam (2015). Another use case is the analysis of open-ended survey responses, as described in Mossholder et al. (1995), to measure the happiness of our society and Desmet and Hoste (2013) propose that it can be used for suicide prevention.

Since emotion is a rather abstract concept, psychologists have developed emotion models which can have varying degrees of complexity and depth. Those models are used to obtain classes of emotions for the machine to match. One model is the basic emotion model described in Ekman and Friesen (1971) and is often used in emotion analysis. A more complex model is the circumplex model proposed by Russell (1980) and improved by Mehrabian (1980) and was used in Zhang et al. (2015) on psychobiological data, like breathing rate. There is by no means a consensus on a "best" emotion model and different models have been criticized for different reasons. For example, Ortony and Turner (1990) question the existence of basic emotions and Mesquita and

Frijda (1992) found out that emotions differ from culture to culture. That means that the emotion model used can vary from research to research and as such many emotion classifiers exist that are based on different models.
In other words, the corpora that a classifier is trained on is annotated on basis of a specific model which means that the classifier can only match emotions to the emotion classes of that model. Also, it is not possible for that classifier to train on different corpora of another model. Although some models have many similarities, other models are fundamentally different which makes a mapping between models not trivial. As an example, a classifier based on Ekman's model does not analyse the intensity of a emotion, whereas a classifier based on a model which also takes account of the intensity of an emotion, like Plutchik's model, will analyse it.
By using a multitask learning method across different emotion models one can circumvent this problem. Multitask learning (MTL) is a method, where the learner tries to optimize multiple similar tasks instead of a single one. MTL is described in detail in Caruana (1997). With MTL one classifier that trains on different emotion models is built. This classifier consists of emotion model specific layers and shared layers. When training on a corpora of a specific model the training feedback not only improves the model specific layer but also the shared layers, which means that classification for the different emotion models is improved as well. Not only can all kinds of different corpora be used now, similarities between the models can be exploited to improve the learning effect by sharing information and feedback. Furthermore, once the classifier is sufficiently trained, a corpora that is annotated on basis of one specific emotion model can be used as input to determine annotations for all other emotion models and expand that corpora such that it covers all learned emotion models.
The multitask learning method may prove useful to create an annotator, which can create annotations for every emotion model given a text input and eventually be able to annotate whole corporas automatically. But it is to be seen, if the multitask learning approach introduces to much noise to the

annotations, or if the multitask classifier can exploit the similarities shared by the tasks and improve the performance in terms of precision and recall. Concretely, the research question is: Are annotations by a multitask learning classifier too noisy or good enough to observe interactions between the models?
Our hypothesis is, that the latter is true and thus the goal of this bachelor thesis is to evaluate the merit of multitask learning across emotion models. Specifically, the F1-score / mean-squared-error of a multitask classifier is to be compared to the singletask classifier.
This thesis is structered as follows: After this introduction, there will be an overview over the related work. There, brief summaries of the psychological emotion models that are used are given. Further, the basics of text classification and its evaluation are formalized. Afterwards, influencial papers on emotion analysis from text are listed and finally relevant papers on MTL are shown. In the next section, our methods are explained, i.e., how our models were build and what parameters were tuned. In section 4, the experiment is presented. This contains what data sets were used in which setting and what result came out. Then the discussion will take place, where the effects of MTL is evaluated and where it is decided to accept or reject our hypothesis. The final section concludes this thesis and there will be an outlook on future works.

2 Related Work

2.1 Emotion models in psychology

As mentioned there are several emotion models that resulted from work of different scientists. The emotion models that will be used for our classifier are as follows:

Six basic Ekman emotions

Ekman and Friesen (1971) researched, if emotion and facial expressions were consistent across cultures. In their study and similar studies of other scientists, they found evidence to support their hypothesis, that there are some discrete emotions that are universal. They postulate that humans have six basic emotions that are consistent across cultures. These emotions are happiness, anger, surprise, fear, sadness and disgust.

Plutchik's wheel of emotions

In Plutchik (1980), he describes a model with eight basic emotions; to the six basic Ekman emotions trust and anticipation are added. The eight emotions are placed on a wheel, where similar emotions are next to each other and opposite emotions like joy and sadness are opposite to each other. Further each emotion has an intensity, with the more intense emotion in the inner part while the less intense emotions are on the outer part of the wheel, e.g., annoyance to anger to rage. Also, there are emotions in between two neighbouring basic emotion which indicates a mixture of these emotion, for example love is a mixture of joy and trust.

Circumplex model by Russell (1980)

This model maps emotions on a two-dimensional plane, with valence and arousal as dimensions. Russel postulates that emotion can be represented by these two quantities. For example, sleepiness has a very negativ arousal value and a near neutral valence value while excitement has a high arousal value and a positiv valence value. Later, in Mehrabian (1980), a third dimension dominance was added which represented how much the emotion is controlling the feeler.

Patterns of cognitive appraisal by Smith and Ellsworth (1985)

Smith and Ellsworth use different dimensions of cognitive appraisal to parameterise emotions. Cognitive appraisal refers to Scherer (1982), specifically

his Component Process Model. The dimensions are: pleasentness, responsibility/control, certainty, attention, effort and situational control.

2.2 Text classification

Classification is a method that assigns labels or classes to object representations. In text classification these object representations can be phrases, titles or other bodies of text. As a general term, they are called documents.
We define these documents as $x_i \in X$ (X is the document space), which are to be mapped to labels $Y = \{y_1, ..., y_k\}$. In supervised learning, a data set with correctly annotated instances is given, with the goal to train a classifier that can then classify unseen data correctly. A classifier is a function $f : X \rightarrow Y$ and the objective of machine learning methods is to learn a function f such that the error is minimized.
While this thesis addresses different emotions as labels, they can be many different things, like language identification (Lui and Baldwin (2012)), topic labeling (Sriram et al. (2010)), spam filtering (Cohen et al. (1996)) and others.
Theoretically, all classification algorithms for classification can be used in text classification, but due to certain characteristics that text classification has, e.g., high dimensional and sparse feature spaces, some algorithms are to be preferred. A survey on text classification algortihms by Aggarwal and Zhai (2012) found out that most algortihms for classification like decision trees, rules, Bayes methods, nearest neighbor classifiers, SVM classifiers, and neural networks were adapted and used for text classification. In particular neural networks and SVM classifiers, which seem to work well with the aforementioned charecteristics, were researched. SVMs or Support-Vector-Machines were first proposed in Cortes and Vapnik (1995) and one of the first implementations for text classification was in Joachims (1998). More recently, neural networks based on long short-term memory (LSTM) or recurrent convolutional neural networks (R-CNN) have shown great performance, like in

Lai et al. (2015).

2.3 Emotion analysis from text

Now, we try to give a comprehensive, albeit not complete, overview over the various works in this field. For more details we refer to the several surveys on Emotion Analysis like Hakak et al. (2017) or Kim and Klinger (2019).
As mentioned in the introduction, many emotion classifiers exist that are based on different models. Early works on emotion analysis from text like Alm et al. (2005) or Strapparava and Mihalcea (2008) used a text-based emotion classifier to match text to the basic emotions in Ekman's model, but that does not mean that more recent work do not use his model, as Li et al. (2017) shows, where a emotion labelled dialog dataset was created.
Mohammad and Kiritchenko (2015), Suttles and Ide (2013) and Meo and Sulis (2017) are examples were Plutchik's model is used to analyse tweets.
It seems that these discrete emotion models are more frequently used than the other two models, perhaps due to their simpler structure. Still, the circumplex model is used in Hasan et al. (2014), Buechel and Hahn (2017) and several other works. The model of Smith and Ellsworth however did not find any application in emotion analysis from text so far and is currently worked with in an unpublished paper in our work group.
As for the classifier, there are many possible approaches. The Support-Vector-Machine (SVM) is one popular approach that is used in Roberts et al. (2012) or Zehe et al. (2016). Yu (2008) compared SVM to a Naive Bayes approach. Maximum Entropy classifier are used in Rao et al. (2016) and Wicentowski and Sydes (2012), while using decision trees is also a possibility, as Samothrakis and Fasli (2015) did. There are many more approaches like unsupervised learning or lexicon based approaches that have been trialed before and it is also possible to build multiple classifiers and determine the best performing one, as in Perikos and Hatzilygeroudis (2016), Suttles and Ide (2013) or Hasan et al. (2014).

For neural networks, Kim (2014) employed a convolutional neural network (CNN) with pre-trained word vectors to classify sentiment on sentence level. Su et al. (2018) proposes that using a long-short term memory (LSTM)-based approach yields is an improvement over a CNN-based approach.
For the examined text corpora, there are a variety of possibilities: tweets (Roberts et al. (2012), Suttles and Ide (2013)), novels (Mohammad (2013)), news headlines (Strapparava and Mihalcea (2008)) and even suicide notes (Wicentowski and Sydes (2012)) were analysed.
A problem however is that different annotated corpora not only differ in the underlying emotion model but also in their structure. Work to unify a collection of corpora to a common file format with a common annotation schema has already been done in Bostan and Klinger (2018) which is essential to this research since we will use the unified corpora in our research.

2.4 Multitask learning

We speak of Multi Task Learning, when a machine learning model is trained on more than one task. MTL is described in Caruana (1997). There, not only a motivation and possible applications are given, but also the core mechanisms of MTL that explain why it works. These are:

- Statistical Data Amplification: Because of the increase in sample size due the multiple tasks and added individual noises of other tasks, each tasks can perform better
- Attribute Selection: Directly as a consequence of statistical data amplification, the MTL model can now better distinguish between relevant and irrelevant features
- Eavesdropping: Tasks that have difficulty learning a certain feature can eavesdrop on task, for which that feature is easy to learn, which helps the learning of that feature

- Representation Bias: MTL biases the model to prefer representations that other tasks also prefer and avoid representations that other tasks also avoid

According to Ruder (2017) MTL has been used successfully in many domains already, like natural language processing, speech recognition (Deng et al. (2013)), computer vision (Girshick (2015)) and drug discovery (Ramsundar et al. (2015)). Further, they state that there are two main methods for MTL in context of deep learning: hard and soft parameter tuning. For hard parameter tuning, there are shared hidden layers between all taks with several indivdual singletask layers on top. With soft parameter tuning, instead of shared parameters, there are task specific parameters that are constrained in such a way, so that they still are similar. Hard parameter sharing is advantageous if the tasks are closely related, if not, then soft parameter tuning is more beneficial, where the model can also learn what to share.
MTL can also be applied in non-neural models like linear models, kernel methods, and Bayesian algorithms but most commonly Deep Learning is used as a model.
Collobert and Weston (2008) and Liu et al. (2016) used MTL for text classification. They proposed different MTL-architectures for natural language processing and showed the effectiveness of the MTL-classifier on several text-classification tasks, compared to STL-classifier. However, the usage of MTL across different emotion models has, as far as we know, not been done before.

3 Methods

In the following section an overview over the learning models that are used and how they were build is given. In general, our models are fed with features and are expected a prediction, which depending on the data set is a singular emotion, a set of emotion or scalar values for different attribute dimensions.

3.1 Data set splitting

To train and evaluate the classifiers, the data set is often split into a training set and a test set. In the learning process, the classifier receives input instances from the training set and compares the prediction to the actual label. The training set is randomly split in two parts, where one part is trained upon while the other part is a validation set, where the current performance of the classifier can be meassured and if necessary parameters are tuned. Once confident in the classifier, its performance is measured on the test set, which contains never seen before instances.

3.2 Feature extraction

To make documents comprehensive for most algorithms, they need to be represented by a feature vector, i.e., an n-dimensional vector which contains numerical representations of an object. There are different methods to vectorize a document, one is to encode each occurence of a term in a binary vector. In this case the dimension of the vector equals the size of the vocabulary, i.e., the amount of unique words in the training set. Alternativly, one can use term frequencies or something more sophisticated like tf-idf values, which is one of the methods that were used in this work. Tf-idf weighting is a weighting scheme that assigns higher weights to less common words and lower weights to common words, that often have less importance for classification. The formular for the tf-idf value for a term t in a document d is:

$$\text{tf-idf}(t, d) = \text{tf}(t, d) \cdot \text{idf}(t)$$

with tf being the term frequency, i.e., the number of occurences of t in d and idf being the inverse document frequency, which is calculated as:

$$\text{idf}(t) = \log \frac{N}{\Sigma_{D:t \in D} 1}$$

N is the amount of documents in the set. In the scikit learn vectorizer that is used here, the idf-value is also smoothed:

$$\text{idf}(t) = \log \frac{1+N}{1+\Sigma_{D:t\in D}1} + 1$$

There are also several preprocessing methods like stemming, lemmatization and stop words removal to reduce the amount of features and increase performance.

3.3 Singletask neural networks

3.3.1 SVMs and other learning methods

To get an estimate on how regular, out of the box classifiers perform, we use the scikit-learn library to build and train singletask classifier for each emotion model. Commonly used learning methods were used, i.e., Naive Bayes, Support Vector Machines, maximimum Entropy and Perceptron. As for the features, the TfidfVectorizer class was used to extract the tf-idf values from the documents. The classifiers were then fitted with the feature vectors and the annotations. Once trained, those classifiers were able to predict the labels given the feature vectors of the test set.

3.3.2 Neural networks

The idea was to build the multitask classifier out of the singletask classifier. To do that, the singletask classifier had to have a compatible architecture with the structure of the multitask classifier. It was decided that the multitask classifier would be a multiheaded neural network, and so it made sense to build singletask classifiers using neural networks. For that there are two main libraries in python: keras and pytorch. The latter was used to build the models, since it offered more flexibility.
For the uninitiated, a short description for neural networks is given. A neural network typically consists of an input layer, one or more hidden layers and

an output layer. Each layer consists of neurons, and each neuron receives a set of input values x_i. A neuron has certain weights w_i, assigned to each of its inputs and a bias value b. Normally, every neuron computes its output as:

$$f(x) = b + \Sigma w_i \cdot x_i$$

This output is then used as input for the neurons in the next layer, repeating until the output layer. Figure 1 shows the architecture of a fully connected neural network with one hidden layer. Another feature of neural networks are

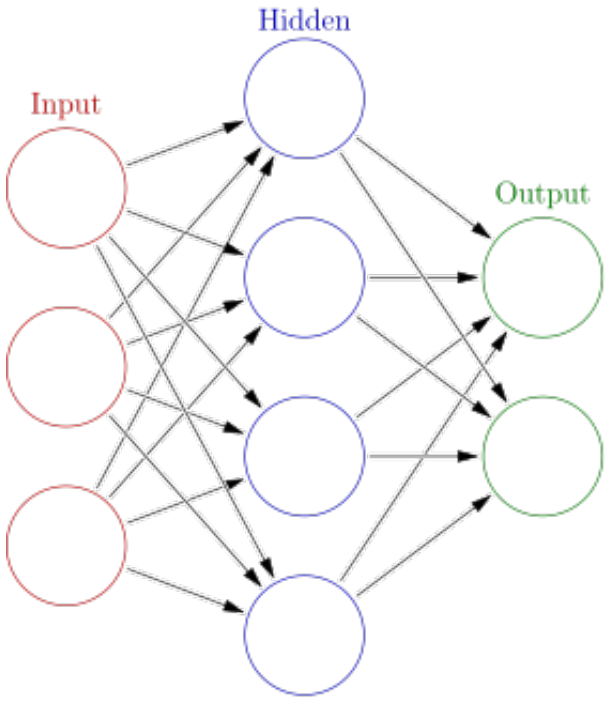

Figure 1: Architecture of a fully connected neural network

their activation functions. These are non-linear functions that are applied on each output, so that the neural network can learn non-linear correlations. Commonly used activation functions are ReLu and sigmoid:

$$\text{ReLu}(x) = \max(x, 0)$$

$$\text{sigmoid}(x) = \frac{e^x}{e^x + 1}$$

Supervised learning in neural networks works as follows: In classification, a loss function is a way to measure the performance of a classifier. It compares the predicted labels with the actual labels and tries to minimize its loss. In neural networks, upon predicting a batch of labels, the loss is calculated and the weight and biases are changed to reduce the loss. This is done by back-propagation, where the gradient of the loss function is calculated to get the direction of change for the parameters of the neural network. This optimization step can be done via algortihms like stochastic gradient descent.
The pytorch library comes with many different loss functions and optimizers. For regression problems like VAD, MSELoss is used while for multiclass problems like Ekman NLLLoss is used and for multilabel problem BCELoss is used.

3.3.3 Embeddings and LSTMs

Another way to extract features in text classification is to use pre-trained embeddings. Here, each word is represented by a vector of fixed size d, such that semantically similar words are closer in the d-dimensional vector space. To use embeddings, the documents is reduced to a bag-of-word representation, i.e., a set of token-ids and is given to a neural network, which has an embedding layer. There, each token-id is turned into the corresponding word vector, using the embedded weight matrix with which the embedding layer was initialized. The advantage of embeddings is that, compared to the sparse high dimensional feature space when using tf-idf values, its feature space is denser and has lower dimensionality, which contributes to its performance.
The main motivator for using embeddings however, is that it works well with a LSTM architecture. LSTM stands for Long Short-Term Memory and was first proposed in Hochreiter and Schmidhuber (1997). In contrast to other recurrent neural networks (RNNs), which neurons are connected along a temporal sequence, enabling the propagation of historical information, LSTMs can retain the information over a long period of time. As such, by using embeddings, where each position of a value in a word vector is correlated

with the same position of the next word vectors, the LSTM can learn that correlation, resulting in better performance.

3.3.4 Overall structure

The overall structure of our neural networks were all similar with minor differences due to the amount of emotions in the model and the amount of emotions to predict. An exception is the model for VAD, which unlike the other models is a linear regressor and not a classifier. It is assumed that every model gets the features and annotations of the training and validation set as input. Three different types of neural networks were implemented for each emotion model: one with tf-idf values as features, one which used pre-trained embedding and one which also used those embeddings but was a LSTM. The ouput vector, contains a value for each possible emotion in the psycological model.

For the first type, a simple neural network with one fully connected linear layer was build. The input dimension was equal to the number of features and the output dimension was the amount of classes of the underlying emotion model. The weights of the layer are randomized on initialization. For the Ekman classifiers, the output went through softmax function, so that classifier can pick the emotion with the highest outgoing value as prediction. For all other classifiers, barring the VAD one, a sigmoid function is used instead. Using the sigmoid function, the outgoing values can be interpreted as probabilities for certain emotions. Here, the classifier predicts all emotions with probabilty greater than fifty percent.

For the second type, an embedding layer is simply added before the linear layer. The weights of this embedding layer is initialized with the weights from the GloVe vectors (Pennington et al. (2014)), that are used as pre-trained embeddings. Those embeddings were trained on about two billion tweets and the dimension size is 200. The features in this setting are token IDs. Each token ID correponds to a row in the pre-trained weight table and after passing through the embedding layer, the one dimensional list of token IDs is

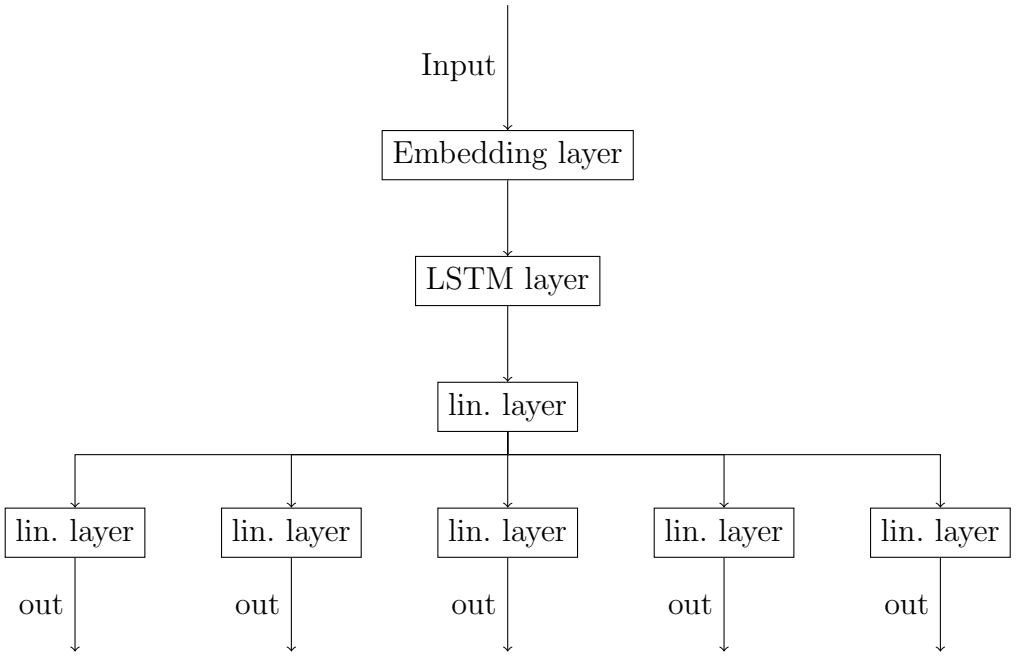

Figure 2: Architecture of the multitask neural network

now a two dimensional list of the corresponding GloVe vectors. This list is then flattened, i.e., transformed to a one dimensional list by concatenating the GloVe vectors and put into the linear layer, which once again outputs the probabilities for the different classes.
The third type is similar to the previous type, but this time there is a LSTM layer after the embedding layer and before the flattening.

3.4 Multitask neural network

The architecture of the multitask model is a multiheaded neural network. It consists of shared layers, through which all inputs go and multiple heads, i.e., multiple task specific layers, that each produce an output. The shared layer is a linear layer, if embeddings are used, an embedding layer and an LSTM layer preceeds it. Each head consists of a activation function and a linear layer, with a fitting activation function for the output, i.e., softmax for

multiclass problems, sigmoid for multilabel problems and none for regression problems. In figure 2 the architecture of a 5-headed neural network is illustrated.
The general idea behind multiheaded neural networks is that the shared layer is learning patterns that are relevant for every head, while each head is learning their specific task only. As a result, for each input multiple outputs are obtained, one for each head. Each head also has a specific loss function, which yields one loss per head. Depending on the training instance, the respective loss is used in backpropagation. When inputting first data set the shared layer and the first head is trained. The next data sets now have the advantage of already trained shared layers, which should increase their performance. Training multiple data sets also makes sure, that the shared layers are not overfitted and can generalize well, which should be beneficial for predicting the unseen test instances later in the testing phase.
For the training set / validation set split, a randomized 80-20 split was used, since k-fold cross-validation would only extend the already long training process.

3.5 Evaluation

One common evaluation metric for classification tasks is the f1-score. The f1-score usually is the harmonic mean of the precision p and recall r:

$$F_1 = 2 \cdot \frac{p \cdot r}{p + r}$$

Precision and recall are defined for each label l as:

$$p_l = \frac{tp}{tp + fp} \qquad r_l = \frac{tp}{tp + fn}$$

tp is the amount of instances, where the label was true and predicted, fp is the amount of instances, where the the label was predicted but not true and fn is the amount of instances, where the label was true but not predicted. Those label dependent precision and recall values can be macro-averaged by

computing the f1-score from their means or they can be micro-averaged by computing label dependent f1-scores and then taking the mean score.

4 Experiment

4.1 Data sets

Annotated corpora for each emotion model will be taken from Bostan and Klinger (2018). There, existing datasets of emotion-annotated texts were collected and put in a unified format. The data sets range from different sources and subjects and use different emotion models for the annotation. Concretly, the data sets are: TEC (Twitter Emotion Corpus by Mohammad (2012)) which consists of over 20,000 tweets annotated corresponding to the six basic Ekman emotion model, SSEC (Stance Sentiment Emotion Corpus by Schuff et al. (2017)) which was also consists of almost 5,000 tweets but was annotated using Plutchik's emotion model, EmoBank (by Buechel and Hahn (2017)) which consists of over 10,000 sentences, annotated according to the valence-arousal-dominance model and ISEAR (International Survey On Emotion Antecedents And Reactions by Scherer and Wallbott (1994)), consisting of 7,665 questionaire answers labelled with the six basic Ekman emotions extended with the emotions shame and guilt. The corpora, annotated on the emotion model by Smith and Ellsworth, containing about 1,000 sentences, is provided from another work in our work group: Hofmann et al. (2020). The data sets are to be divided into a training set and a test set. Data sets, that were not already divided, were split randomly. The training set and the test set are pairwise disjoint. During developement the training was further divided into training and validation. Once the hyperparameters were optimized, the results were measured on the test set. For the MTL classifier, the union of all data sets was used.

4.2 Experimental settings

There are different experimental settings that were used. The first setting that could be set was the different mode of feature extraction. Some classifiers used tf-idf values as features while the other possibility was the use of pre-trained embeddings. As mentioned, the weights from the GloVe vectors Pennington et al. (2014) were used. In particular, the GloVe vectors, which were obtained from about 2 billion tweets, were used. The next experimental setting was the model of the classifier. One possibility was the SVM from the scikit-learn library, which only works with tf-idf values as features. Another posssibility was the singletask neural network that was built specifically for each data set and the last option was the multitask neural network with the multihead architecture. If the latter case was chosen, one had to decide on a third experimental setting, that is which data sets to train the multiheaded neural network on. Training on a data set, would train the shared layer and its respective head. Since there are five data sets, there are 32 possible combinations of chosen data sets to train from, including the case where none are trained. For this experiment, the combinations were limited to training on exactly one, two, or all five data sets. Training on only one data set, makes that effectivly a singletask classifier. Also, to test the classifier on a certain data set, it has to be trained on at least said data set. That means that the multihead classifiers in the result table for the TEC data set were all trained on at least the TEC data set. The same holds for all other data sets.
Knowing this, the resulting tables can be explained as follows: the tables 1 to 5 show the measured results of the respective data sets. In each table the f1-score is shown, where each columns represents a label and the last two columns show the micro and macro average. Each row represents a classifier with a specific experimental setting. The classifiers are split into two sets, seperated by a horizontal line. The classifiers in the first set use the tfidf-values, while the classifiers in the second set use embeddings. In both sets the first rows are the singletask classifiers: the SVM (only in the first set), the singletask neural network (singletask) and the multihead classifiers

	anger	disgust	fear	joy	sadness	surprise	micro average	macro average
SVM	0.39	0.21	0.62	0.71	0.49	0.54	0.49	0.52
singletask	0.37	0.3	0.61	0.7	0.46	0.52	0.49	0.5
1-head	0.37	0.25	0.6	0.66	0.46	0.52	0.48	0.48
5-head	0.35	0.26	0.59	0.67	0.43	0.52	0.47	0.47
+SSEC	0.36	0.28	0.62	0.69	0.46	0.52	0.49	0.49
+ISEAR	0.39	0.28	0.62	0.69	0.43	0.52	0.49	0.49
+S&E	0.37	0.25	0.61	0.7	0.43	0.51	0.48	0.48
+Emobank	0.35	0.27	0.61	0.68	0.45	0.52	0.48	0.49
singletask	0.25	0.11	0.5	0.66	0.4	0.44	0.39	0.4
1-head	0.22	0.11	0.47	0.63	0.36	0.42	0.37	0.38
5-head	0.26	0.09	0.49	0.62	0.26	0.34	0.34	0.37
+SSEC	0.26	0.14	0.46	0.65	0.38	0.44	0.39	0.39
+ISEAR	0.26	0.17	0.51	0.59	0.41	0.41	0.39	0.4
+S&E	0.27	0.14	0.48	0.63	0.39	0.45	0.39	0.4
+Emobank	0.27	0.12	0.47	0.63	0.38	0.41	0.38	0.39

Table 1: f1-score for TEC data set, the first set of rows are classifiers using tf-idf values as features, the second set of rows are LSTMs with embeddings

trained only on the tested data set (1-head). The next five rows represent the multitask classifiers: one trained on all data sets (5-heads) and four trained on two data sets: the tested data set and another secondary data set (+data set).

4.3 Results

Looking at the table 1, the f1-scores for our singletask neural network, using tf-idf values as features, performs best. It also compares relativly well to

	anger	anticipation	disgust	fear	joy	sadness	surprise	trust	micro average	macro average
SVM	0.58	0.57	0.69	0.75	0.75	0.53	0.84	0.81	0.69	0.69
singletask	0.59	0.57	0.67	0.73	0.74	0.56	0.84	0.8	0.69	0.69
1-head	0.49	0.52	0.7	0.74	0.77	0.46	0.84	0.8	0.66	0.69
5-head	0.59	0.44	0.7	0.73	0.74	0.56	0.84	0.79	0.67	0.68
+TEC	0.57	0.47	0.69	0.74	0.74	0.53	0.84	0.78	0.67	0.68
+ISEAR	0.57	0.49	0.68	0.74	0.72	0.51	0.84	0.8	0.67	0.68
+S&E	0.46	0.32	0.7	0.74	0.76	0.35	0.84	0.8	0.62	0.67
+Emobank	0.29	0.25	0.71	0.74	0.78	0.29	0.84	0.79	0.59	0.67
singletask	0.56	0.36	0.69	0.7	0.68	0.53	0.84	0.74	0.64	0.64
1-head	0.61	0.56	0.68	0.69	0.72	0.56	0.83	0.76	0.68	0.68
5-head	0.58	0.46	0.67	0.72	0.68	0.56	0.84	0.74	0.65	0.66
+TEC	0.57	0.46	0.7	0.74	0.72	0.53	0.84	0.78	0.67	0.67
+ISEAR	0.56	0.36	0.69	0.7	0.68	0.53	0.84	0.74	0.64	0.64
+S&E	0.6	0.54	0.65	0.72	0.71	0.57	0.83	0.76	0.67	0.68
+Emobank	0	0	0.7	0.74	0.76	0	0.84	0.79	0.48	0.48

Table 2: f1-score for SSEC data set, the first set of rows are classifiers using tf-idf values as features, the second set of rows are LSTMs with embeddings

other models, like the SVM classifier which had an f1-score of 0.49 (micro) and 0.52 (macro) or the one in Bostan and Klinger (2018), which had a micro averaged f1-score of 0.48 with the macro average being the same. For the LSTMs, they are about 0.1 lower than their counterparts. In both cases however, slight improvements can be noticed when training is done on two data sets compared to training only on one data set in the multihead network. Across the labels, no remarkable outliers can be seen.

In table 2 are the results for the SSEC data set, which was annotated using Plutchik's wheel of emotions. All of the singleask classifier including the SVM

	anger	disgust	fear	guilt	joy	sadness	shame	surprise	micro average	macro average
SVM	0.46	0.59	0.71	0.46	0.71	0.63	0.51	0	0.51	0.51
singletask	0.4	0.59	0.68	0.45	0.66	0.57	0.51	0	0.48	0.48
1-head	0.39	0.59	0.67	0.42	0.67	0.55	0.5	0	0.39	0.43
5-head	0.39	0.55	0.69	0.44	0.68	0.58	0.5	0	0.4	0.44
+TEC	0.36	0.61	0.67	0.45	0.66	0.55	0.51	0	0.39	0.43
+SSEC	0.35	0.57	0.67	0.43	0.68	0.57	0.5	0	0.39	0.43
+S&E	0.42	0.56	0.67	0.42	0.65	0.56	0.49	0	0.39	0.43
+Emobank	0.36	0.59	0.65	0.44	0.67	0.6	0.5	0	0.39	0.43
singletask	0.22	0.43	0.63	0.37	0.57	0.45	0.26	0	0.37	0.37
1-head	0.26	0.39	0.52	0.28	0.48	0.4	0.29	0	0.26	0.29
5-head	0.25	0.3	0.44	0.19	0.48	0.41	0.15	0	0.23	0.27
+TEC	0.29	0.32	0.6	0.25	0.59	0.44	0.31	0	0.28	0.31
+SSEC	0.34	0.34	0.53	0.32	0.56	0.42	0.32	0	0.29	0.33
+S&E	0.24	0.36	0.56	0.33	0.5	0.45	0.34	0	0.29	0.32
+Emobank	0.25	0.41	0.59	0.29	0.51	0.46	0.24	0	0.27	0.3

Table 3: f1-score for ISEAR data set, the first set of rows are classifiers using tf-idf values as features, the second set of rows are LSTMs with embeddings

perform similar across the board. Compared to Schuff et al. (2017), one can see that the results are an improvement. Here, the multitask classifiers do not have an advantage over the singletask classifiers. Intersestingly, a drop in th f1-score for the label anticipation can be seen in the multitask models. While all other labels have comparable f1-scores, the 1-head score for anticipation is usually higher than the other classifiers.

Another data set is the ISEAR data set, which expanded Ekman's six basic emotions by guilt and shame. Comparing the results of the SVM and singletask neural networks with tf-idf values as features to the results in

	attention	certainty	circumstance	control	effort	pleasant	responsibility	micro average	macro average
SVM	0.82	0.9	0.68	0.4	0.57	0.36	0.44	0.6	0.64
singletask	0.82	0.88	0.69	0.4	0.56	0.43	0.43	0.6	0.64
1-head	0.75	0.89	0	0	0	0	0	0.23	0.23
5-head	0.75	0.89	0.48	0.08	0.27	0	0.11	0.37	0.49
+TEC	0.75	0.89	0	0	0	0	0	0.23	0.23
+SSEC	0.75	0.89	0	0	0	0	0	0.23	0.23
+ISEAR	0.76	0.89	0.52	0.08	0.33	0	0.14	0.4	0.5
+Emobank	0.75	0.89	0	0	0	0	0	0.23	0.23
singletask	0.77	0.87	0.56	0.26	0.4	0.27	0.47	0.52	0.53
1-head	0.71	0.88	0.53	0.59	0.57	0.39	0.35	0.58	0.61
5-head	0.76	0.89	0.46	0.41	0.03	0	0.3	0.41	0.46
+TEC	0.75	0.89	0.3	0.08	0.16	0	0	0.31	0.36
+SSEC	0.8	0.85	0.5	0.44	0.52	0.25	0.47	0.55	0.56
+ISEAR	0.78	0.88	0.38	0.52	0.29	0.06	0.18	0.44	0.5
+Emobank	0.75	0.89	0	0	0	0	0	0.23	0.23

Table 4: f1-score for Smith & Ellsworth data set, the first set of rows are classifiers using tf-idf values as features, the second set of rows are LSTMs with embeddings

Bostan and Klinger (2018), they seem on the same level. The LSTMs with embeddings are performing worse. Some improvements can be seen in a few multitask classifiers compared to their 1-head counterpart. As for the labels, no striking differences between the models can be seen.

For the Smith & Ellsworth data set, the only other results are in Hofmann et al. (2020), which achieved a slightly better score than our top models and our SVM, with a macro averaged f1-score of 70 and a micro averaged f1-score of 75. Conspicuous are the bad results when using the tf-idf features

	arousal	dominance	valence	average
SVM	0.08	0.06	0.04	0.06
singletask	0.11	0.06	0.05	0.07
1-head	0.09	0.07	0.05	0.07
5-head	0.1	0.08	0.05	0.08
+TEC	0.1	0.07	0.05	0.07
+SSEC	0.09	0.07	0.05	0.07
+ISEAR	0.1	0.07	0.05	0.07
+S&E	0.09	0.07	0.05	0.07
singletask	0.13	0.07	0.05	0.08
1-head	0.09	0.06	0.04	0.07
5-head	0.11	0.06	0.04	0.07
+TEC	0.11	0.06	0.04	0.07
+SSEC	0.12	0.06	0.04	0.08
+ISEAR	0.11	0.06	0.04	0.07
+S&E	0.12	0.06	0.04	0.07

Table 5: MSE for Smith & Ellsworth data set, the first set of rows are classifiers using tf-idf values as features, the second set of rows are LSTMs with embeddings

in the multihead network, causing many zeros in the less frequent occuring labels. While using all five heads improves the results, so does the addition of ISEAR.

Lastly, the results for the EmoBank data set: table 5 shows the mean-squared-error (MSE). The labels were in range of the intervall $[1, 5]$. No noticeable differences can be seen across the table.

5 Discussion

Overall, the performance of our singletask classifiers compare to other recent classifiers and out of the box classifiers like the SVMs from scikit-learn. None of our multitask classifiers however, could surpass the performance of those models.

5.1 Effect of multitask classification

In most cases, training on all data sets leads to worse performance per data set, than training only on that data set. The only two exception can be observed in the classifiers using tf-idf features of the ISEAR and the Smith & Ellsworth data set. In those cases, the 5-heads classifier outperforms the 1-head classifier. It has to be noted though, that the 1-head classifier of the Smith & Ellsworth data set performs subpar and upon closer inspection never predicted five out of the seven labels. On that note, one might conjecture that the additional training of the 5-heads classifier boosted the learning effect in this case. Yet in all other cases, the 5-heads classifier performed slightly worse than its 1-head classifier counterpart. This suggests that training on to many data sets is detrimental to the performance. The fact that most settings with two trained heads outperform the setting with all heads trained further support the theory, that training on all five data sets introduces too much noise for any improvements.
Looking at the 2-headed classifiers (represented in the rows with +data set), the results vary very much. In the TEC data set slight improvements for almost all 2-headed classifiers compared to the 1-head classifier can be seen. The results are also very close to the performance of the singletask classifier, albeit not better. The TEC data set is the biggest data set of all five data sets though, and thus, it is possible that the effects of multitask learning is very subdued. Looking at the SSEC data set, which is one of the smaller data set, all 2-headed classifiers have worse results, especially the SSEC+Emobank

combination, while the SSEC+TEC combination performs the best. In the ISEAR data set, the 2-headed classifiers perform on par with the 1-head classifier. Surprisingly, all combinations perform very similary. For the Smith & Ellsworth data set, there is an improvement in the ISEAR row, but again, that is only because the 1-head classifier using tfi-idf features performs subpar. Noticeable, are the many zeroes, indicating that the classifiers do not sufficiently learn, perhaps due to the small training set of less than 1,000 instances. Regarding the regressor of the Emobank data set, the results of the multitask classifiers do not improve the performance. Also, out of all 2-headed classifiers, those with +Emobank perform among the worse. It seems that the addition of the Emobank data set as a secondary training set is disadvantageous for the performance of the classifier. The reason for that is likely because the tasks to be learned are to different, for any synergies to show up. Unlike all other data set, which use classifiers to predict the right annotation, Emobank is annotated using the valence-arousal-dominance model, which requires regression.
Looking at the individual labels, each classifier performed almost always similar well on a specific label compared to the other classifiers. There was a case in the SSEC data set where the multitask classifiers performed worse on the label anticipation, perhaps because that label is unique to the SSEC data set. But since the performance on the label trust, which is also unique to SSEC, and the same for the labels guilt and shame, which are also unique, does not show that pattern, the reason might be something else.
Lastly, the effects of multitask classification manifest, disregarding the choice of feature extraction. There was no case, where the multiheaded approach had a different effect depending on the choice of tf-idf values or embeddings.

5.2 Hypothesis acceptance/rejection

Our hypothesis, that the annotations of a multitask learning classifier are good enough to observe interactions between the models, has to be rejected.

In most cases the annotations are to noisy, and a trained multiheaded neural network performs worse than one with only one head is trained and compared to a singletask neural network the performance is even worse. Still, since only a rather simple architecture for multitask learning was used here, a more sophisticated one might lead to better results.

6 Conclusion and Future Work

In summary, multitask classification across psychological models of emotion and affect does not lead to better result, at least not with the multiheaded architecture that was used here. However some improvements over 1-head classifiers could be observered, suggesting that positive effects of multitask classification do exists, but where perhaps held back due to the architecture of the neural network. There might be other MTL methods that can be explored in future works, like cascading features, where the results of one task are the features of the next task.

Another finding is that while the choice of vectorization influenced the overall performance, it did not have an influence on the effect of multitask learning. For the future, different word representations can be used, that are context sensitive, described in Peters et al. (2018). Perhaps those can have a positive effect on multitask learning.

As mentioned in the discussion section, the different sizes of data sets could have been problematic. For example, the TEC+S&E combination consists of about 95% instances from TEC, which makes it too lopsided for any positive multitask learning effects. Perhaps a 50-50 ratio would be ideal, or maybe a main task with a higher ratio and a secondary task with a lower ratio would be better. In a future work, one might try to improve the multiheaded classifier by trying out different ratios and different numbers of heads / layers.

References

Kirk Roberts, Michael A Roach, Joseph Johnson, Josh Guthrie, and Sanda M Harabagiu. Empatweet: Annotating and detecting emotions on twitter. In *Lrec*, volume 12, pages 3806–3813. Citeseer, 2012.

Chew-Yean Yam. Emotion detection and recognition from text using deep learning, 2015. URL `https://www.microsoft.com/developerblog/2015/11/29/emotion-detection-and-recognition-from-text-using-deep-learning/`.

Kevin W. Mossholder, Randall P. Settoon, Stanley G. Harris, and Achilles A. Armenakis. Measuring emotion in open-ended survey responses: An application of textual data analysis. *Journal of Management*, 21(2):335–355, 1995. doi: 10.1177/014920639502100208. URL `https://doi.org/10.1177/014920639502100208`.

Bart Desmet and VéRonique Hoste. Emotion detection in suicide notes. *Expert Systems with Applications*, 40(16):6351–6358, 2013.

Paul Ekman and Wallace V Friesen. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124, 1971.

James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

Albert Mehrabian. *Basic dimensions for a general psychological theory: Implications for personality, social, environmental, and developmental studies*, volume 2. Oelgeschlager, Gunn & Hain Cambridge, MA, 1980.

Lin Zhang, Stefanie Rukavina, Sascha Gruss, Harald C Traue, and Dilana Hazer. Classification analysis for the emotion recognition from psychobiological data. In *ISCT*, pages 149–154, 2015.

Andrew Ortony and Terence Turner. What's basic about basic emotions? *Psychological review*, 97:315–31, 08 1990. doi: 10.1037/0033-295X.97.3.315.

Batja Mesquita and Nico Frijda. Cultural variations in emotions. *Psychological Bulletin*, 112:179–204, 09 1992. doi: 10.1037//0033-2909.112.2.179.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Robert Plutchik. A general psychoevolutionary theory of emotion. In *Theories of emotion*, pages 3–33. Elsevier, 1980.

Craig A Smith and Phoebe C Ellsworth. Patterns of cognitive appraisal in emotion. *Journal of personality and social psychology*, 48(4):813, 1985.

K. Scherer. Emotion as a process: Function, origin and regulation. *Social Science Information*, 21:555 – 570, 1982.

Marco Lui and Timothy Baldwin. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30, 2012.

Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 841–842, 2010.

William W Cohen et al. Learning rules that classify e-mail. In *AAAI spring symposium on machine learning in information access*, volume 18, page 25. Stanford, CA, 1996.

Charu C Aggarwal and ChengXiang Zhai. A survey of text classification algorithms. In *Mining text data*, pages 163–222. Springer, 2012.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

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

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

Nida Hakak, Mohsin Mohd, Mahira Kirmani, and Mudasir mohd. Emotion analysis: A survey. In *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, pages 397–402, 07 2017.

Evgeny Kim and Roman Klinger. A survey on sentiment and emotion analysis for computational literary studies. *Zeitschrift fuer Digitale Geisteswissenschaften*, 4, 2019. doi: http://dx.doi.org/10.17175/2019_008.

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 579–586. Association for Computational Linguistics, 2005.

Carlo Strapparava and Rada Mihalcea. Learning to identify emotions in text. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, page 1556–1560, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781595937537. doi: 10.1145/1363686.1364052. URL `https://doi.org/10.1145/1363686.1364052`.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*, 2017.

Saif M. Mohammad and Svetlana Kiritchenko. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence*, 31(2): 301–326, 2015. doi: 10.1111/coin.12024. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/coin.12024`.

Jared Suttles and Nancy Ide. Distant supervision for emotion classification with discrete binary values. In Alexander Gelbukh, editor, *Computational*

Linguistics and Intelligent Text Processing, pages 121–136, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-37256-8.

Rosa Meo and Emilio Sulis. Processing affect in social media: A comparison of methods to distinguish emotions in tweets. *ACM Transactions on Internet Technology (TOIT)*, 17(1):1–25, 2017.

Maryam Hasan, Elke Rundensteiner, and Emmanuel Agu. Emotex: Detecting emotions in twitter messages. Academy of Science and Engineering (ASE), USA,© ASE 2014, 2014.

Sven Buechel and Udo Hahn. Emobank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 578–585, 2017.

Albin Zehe, Martin Becker, Lena Hettinger, Andreas Hotho, Isabella Reger, and Fotis Jannidis. Prediction of happy endings in german novels based on sentiment information. In *3rd Workshop on Interactions between Data Mining and Natural Language Processing, Riva del Garda, Italy*, 2016.

Bei Yu. An evaluation of text classification methods for literary study. *Literary and Linguistic Computing*, 23(3):327–343, 09 2008. doi: 10.1093/llc/fqn015. URL `https://doi.org/10.1093/llc/fqn015`.

Yanghui Rao, Haoran Xie, Jun Li, Fengmei Jin, Fu Lee Wang, and Qing Li. Social emotion classification of short text via topic-level maximum entropy model. *Information Management*, 53(8):978 – 986, 2016. doi: https://doi.org/10.1016/j.im.2016.04.005. URL `http://www.sciencedirect.com/science/article/pii/S0378720616300386`.

Richard Wicentowski and Matthew R. Sydes. Emotion detection in suicide notes using maximum entropy classification. *Biomedical Informat-*

ics Insights, 5s1:BII.S8972, 2012. doi: 10.4137/BII.S8972. URL https://doi.org/10.4137/BII.S8972.

Spyridon Samothrakis and Maria Fasli. Emotional sentence annotation helps predict fiction genre. *PloS one*, 10(11):e0141922, 2015.

Isidoros Perikos and Ioannis Hatzilygeroudis. Recognizing emotions in text using ensemble of classifiers. *Engineering Applications of Artificial Intelligence*, 51:191 – 201, 2016. doi: https://doi.org/10.1016/j.engappai.2016.01.012. URL http://www.sciencedirect.com/science/article/pii/S0952197616000166. Mining the Humanities: Technologies and Applications.

Yoon Kim. Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. doi: 10.3115/v1/d14-1181. URL http://dx.doi.org/10.3115/v1/D14-1181.

Ming-Hsiang Su, Chung-Hsien Wu, Kun-Yi Huang, and Qian-Bei Hong. Lstm-based text emotion recognition using semantic and emotional word vectors. In *2018 First Asian Conference on Affective Computing and Intelligent Interaction (ACII Asia)*, pages 1–6. IEEE, 2018.

Saif Mohammad. From once upon a time to happily ever after: Tracking emotions in novels and fairy tales. *arXiv preprint arXiv:1309.5909*, 2013.

Laura-Ana-Maria Bostan and Roman Klinger. An analysis of annotated corpora for emotion classification in text. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2104–2119, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1179.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Li Deng, Geoffrey Hinton, and Brian Kingsbury. New types of deep neural network learning for speech recognition and related applications: An overview. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8599–8603. IEEE, 2013.

Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

Bharath Ramsundar, Steven Kearnes, Patrick Riley, Dale Webster, David Konerding, and Vijay Pande. Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072*, 2015.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*, 2016.

scikit learn. Feature extraction. URL `https://scikit-learn.org/stable/modules/feature_extraction.html#tfidf-term-weighting`.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL `http://www.aclweb.org/anthology/D14-1162`.

Saif Mohammad. # emotional tweets. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the*

Sixth International Workshop on Semantic Evaluation (SemEval 2012), pages 246–255, 2012.

Hendrik Schuff, Jeremy Barnes, Julian Mohme, Sebastian Padó, and Roman Klinger. Annotation, modelling and analysis of fine-grained emotions on a stance and sentiment detection corpus. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 13–23, 2017.

Klaus R Scherer and Harald G Wallbott. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology*, 66(2):310, 1994.

Jan Hofmann, Enrica Troiano, Kai Sassenberg, and Roman Klinger. Appraisal theories for emotion classification in text, 2020. URL `https://arxiv.org/abs/2003.14155`.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL `https://www.aclweb.org/anthology/N18-1202`.

Erklärung (Statement of Authorship)

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und dabei keine andere als die angegebene Literatur verwendet habe. Alle Zitate und sinngemäßen Entlehnungen sind als solche unter genauer Angabe der Quelle gekennzeichnet. Die eingereichte Arbeit ist weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen. Sie ist weder vollständig noch in Teilen bereits veröffentlicht. Die beigefügte elektronische Version stimmt mit dem Druckexemplar überein. [1]

(Hai Dang Nguyen)

[1] Non-binding translation for convenience: This text is the result of my own work, and any material from published or unpublished work of others which is used either verbatim or indirectly in the text is credited to the author including details about the exact source in the text. This work has not been part of any other previous examination, neither completely nor in parts. It has neither completeley nor partially been published before. The submitted electronic version is identical to this print version.