AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**

KATEDRA INFORMATYKI STOSOWANEJ

Bachelor of Science Thesis

Metody filtracji strumienia wiadomości pod względem emocjonalnym
Methods of emotional filtering of the message stream

Autor: *Karolina Bogacka*
Kierunek studiów: *Computer Science*
Opiekun pracy: *Krzysztof Kutt PhD, DSc*

Kraków, 2020

Uprzedzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystycznego wykonania albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.", a także uprzedzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust. 1 ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (t.j. Dz. U. z 2012 r. poz. 572, z późn. zm.): „Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchybiające godności studenta student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej «sądem koleżeńskim».", oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i że nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

I hereby thank everyone's patience.

Spis treści

1. Introduction

We live in a challenging and chaotic time.

Especially so for news production and consumption. Based on a survey conducted by the Pew Research Center from February 22nd to March 4th, 2018, almost seven in ten (or, more precisely, 68%) Americans feel worn out by the amount of news available these days, compared to only three in ten reporting contentment with the amount of news they get[1]. As a result, there is a growing number of people simply deciding to avoid this source of information altogether. According to the Reuters Institute's 2017 Digital News Report, between 6 and 57 percent of populations worldwide said they "sometimes" or "often" avoided the news, most often because they did not trust it or found it upsetting[2].

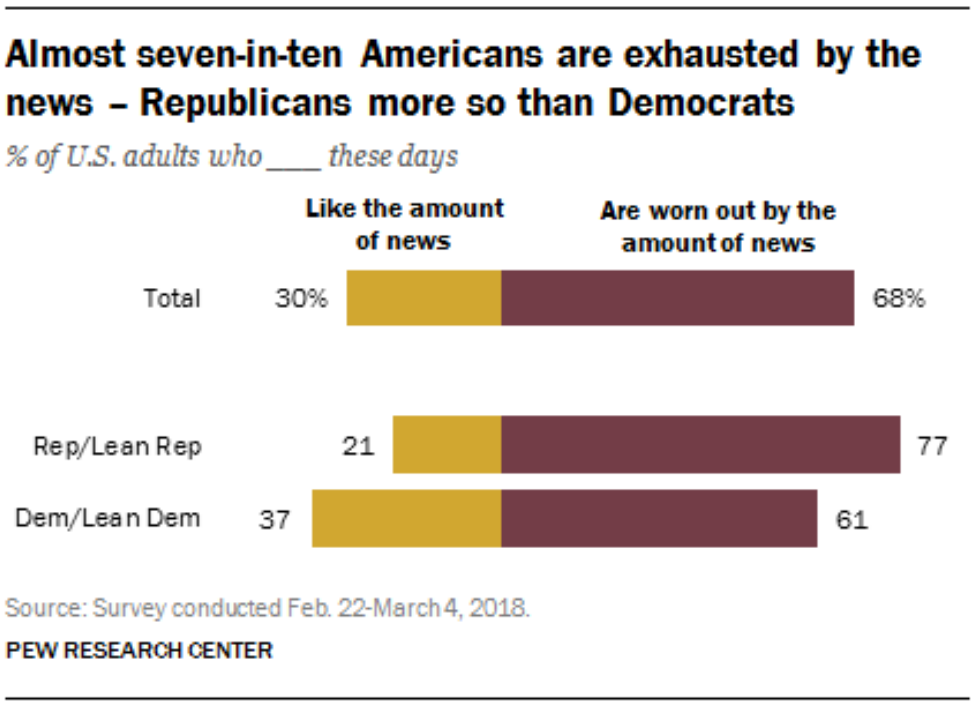
This phenomenon even got its own name - or, rather, got classified as an example of another, more widely known concept - compassion fatigue. Definition of compassion fatigue, formulated by a psychologist, Charles Figley, describes it as "a state of exhaustion and dysfunction, biologically, physiologically and emotionally, as a result of prolonged exposure to compassion stress"[3]. Symptoms, as further described, can be surprisingly diverse. From behavioral changes, such as a sudden lack in the ability to remain objective, including physical (exhaustion, severe anxiety) and emotional changes (numbness, apathy, depression)[4].

And it was not only noticed in psychology. Soon, an analogous concept has emerged in media studies. "It seems as if the media careen from one trauma to another, in a breathless tour of poverty, disease, and death," wrote in her book Susan Moeller, an established journalist and scholar. "The troubles blur. Crises become one crisis."[5] As she observed, exposing oneself to horrific images over and over did not lead to an increase of empathy, but instead to a state similar to one of emotional shutdown. Moreover, the issue seemed to be especially pronounced in the case of news reports.

In Charles Figley's words, "There is a cost to caring"[3].

Moeller's writing proved that compassion fatigue can be a vicious cycle. A cycle emphasized even more by another cycle - the news cycle. And when epidemics and bombings were shown, again and again, they became more and more normalized, even tedious. As was written in The Guardian, "The only way to break through your audience's boredom is to make each disaster feel worse than the last."[4] Which means even more blood, more guilt, more violence, everything to compete with more accessible, local stories.

Media channels are only a regular, non-privileged part of our economic system. To sustain themselves, a significant portion of them has to rely on ad revenue. Their profits are directly correlated to the



Rys. 1.1. Visualisation of the Pew Research Center Survey[1]

amount of time they're able to hold their readers' attention, which often encourages sensationalism and emotional, not informative reporting.

A question remains: do we need to emotionally exhaust ourselves to stay positively involved in the world's affairs, instead of growing disillusioned and numb? According to psychologist Paul Bool, no, we do not. In his mind, the answer is to approach problems more logically, in order to fully retain the clarity of our solutions and observations[6].

1.1. Preordained goals

The goal of this paper is to propose a potential solution to the rise of compassion fatigue among news readers - a news aggregating application, that would allow the reader to filter out articles written in an informative and emotionally bearable manner. As a result, users would be able to personalize the strength of the filter, allowing them to continuously stay informed, even during more stressful periods of their lives.

1.2. Contents

The paper is divided into seven chapters. The first one serves as an introduction to the topic and a curt argument for taking action towards combating the phenomenon of news avoidance. The second chapter introduces the reader to the problem of emotion modeling in text. Chapters three and four cover technical requirements and dataset preparation. The fifth chapter takes the time to precisely explain neural network

architectures, used later in distress modeling. The final aggregator implementation is described in chapter five, chapter six is devoted to project evaluation, and chapter seven to conclusions.

2. Modeling sentiment in text

2.1. Natural Language Processing

Natural Language Processing is a way of computationally processing the human language and producing a correct, socially expected response[7]. The study of Natural Language Processing is divided into two disciplines: written and spoken language processing.

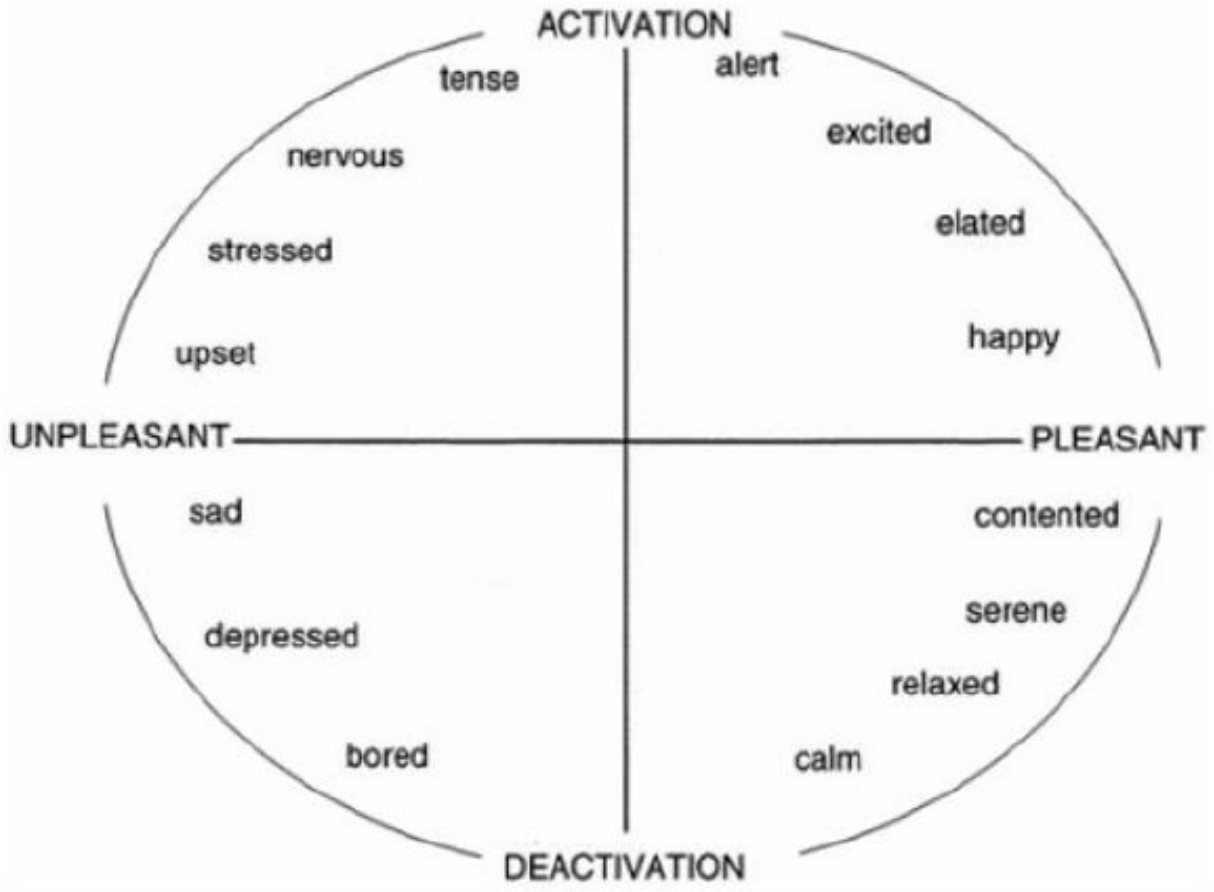
2.2. Sentiment Analysis

Sentiment analysis is a recently popular field of research, which uses Natural Language Processing to recognize and measure emotions in text, with a strong focus on whether the author's expression is negative, positive, or neutral. [8]The reason for this focus can be found in the many business applications of sentiment analysis, from estimating the general opinion about a product from user reviews to recommending further articles to a user based on their reading history.

2.3. Emotion theory

There are many existing models of emotion, some of them more complicated, others - more simplistic. One of the most commonly encountered models, known also as "The Big Six", was used by Ekman in his research on emotion recognition from facial expressions. The list of available emotional states in the model contains happiness, sadness, fear, surprise, anger, and disgust. However, the model assumes these six emotions to be completely separate and it doesn't consider in any way their polarity or intensity[9].

Another model used in sentiment analysis is the circumplex model of affect. It proposes that all affective states arise from cognitive interpretations of core neural sensations, that are the product of two independent neurophysiological systems. "This model stands in contrast to theories of basic emotions, which posit that a discrete and independent neural system subserves every emotion."[10] Its advantages, when it comes to sentiment analysis, lie in its continuity and a two-dimensional representation. Moreover, this model acknowledges the activating or deactivating influence of emotion, which will prove useful later in emotion intensity analysis.



Rys. 2.1. The Circumplex Theory Of Affect[10]

2.4. A review of filtering methods

A valid solution to the problem of measuring emotional intensity in text can be constructed by building an emotional dictionary, as presented by Yanghui Rao and Jingsheng Lei[11]. The algorithm, used to construct the dictionary, uses a method based on topic modeling to connect each word with its emotional meaning. Instead of simply measuring the occurrence of terms purely connected to emotional expressions, such as the word “joy“, the dictionary recognizes the additional encountered in expressions like “bombed a hospital“.

Another useful study, connected to the task of approximating emotions exhibited by the reader, describes a multi-label classification system. It proves to be more suited than the single-label system. However, it does not provide sufficient information connected to intensity measurement[12].

A full, open-source toolkit for emotion recognition in text has been proposed by the scientists from the University of Bari. [13] Unfortunately, the emotion model used by EmoTxt doesn’t seem suitable for intensity measurement. Nevertheless, EmoTxt may prove to be a useful tool in further testing and training of the data.

An interesting solution has also been presented by the name of ASNA, which stands for “An Intelligent Agent for Retrieving and Classifying News on the Basis of Emotion-Affinit“[14]. The agent allows the user to use a news filter based not on topics, but specific emotions. However, it doesn’t recognize the intensity of presented emotions or compares them in a meaningful way.

3. Technical requirements

3.1. News aggregators

News aggregators are programs that (in general) do not produce content on their own. Instead, they use curators and RSS feeds to gather news articles, and then filter and present the most interesting examples. As a rule, news aggregators usually display their findings with full attribution and a link to the original source.

3.2. Rich Site Summary (RSS)

RSS - known originally as “RDF Site Summary“ and eventually split into “Rich Site Summary“ and “Really Simple Syndication“ - a type of web feed. RSS feeds consolidate information sources in one place and provides updates whenever a site (a blog or a news site) publishes new content. Evident popularity of the RSS format can be noticed in the presence of feed add-ons in all major web browsers and numerous online aggregators and readers [15].

3.3. Comparison of news aggregators

– Feedly

  A common aggregator existing from 2008, Feedly allows for news filtering based on a certain event (the Mute Filters), as well as creating a temporary filter or filtering out a specific phrase.

– Google News

  A fairly popular and established aggregator, Google News has been publicly available since February 2002. It’s available as an app both on Android and iOS, and also used on default to enhance Google’s regular search engine. Google News offers news filtering based on the topic (Business, Science, Entertainment, Sport) and the scope (Local News, World). Google News filtering is based on the user’s research history, but can also be modified, albeit in a limited fashion.

– Newspy

Newspy is a news aggregator with customizable searching filters. It allows its users to define sources that will be checked for news (which include RSS channels, Twitter timelines, Reddit posts, and web pages). Then, the user can create one or more rules, that will be used to recognize and prioritize relevant information. Newspy is available as an application on the Google Play Store. [16]

3.4. Tool necessary in news aggregator development

– Python

A high-level language created by Guido van Rossum. It was first released in 1991 and, with its design emphasizing code readability and general purposefulness, quickly became useful to many. Nowadays it's commonly encountered in scientists' computations as well as budding programmers' exercises. Because of its vast standard library, Python is often described as a "batteries included" language. [17]

– Flask

An open-source Python microframework (a framework classified as such, because it doesn't require particular tools or libraries). Instead, Flask supports the use of extensions, that allow the use of object-relational mappers, form validation or upload handling. Flask has been used by both Linkedin and Pinterest.

– MySQL

A free, open-source relational database management system, used by many database-driven web applications, such as Drupal, Joomla, phpBB and Wordpress. It's also used by popular websites like Facebook, Flickr, MediaWiki, Twitter, and YouTube.

– MariaDB

One of the most popular open-source relational databases. Its authors are the original developers of MySQL and it is guaranteed to stay open-source.[18] MariaDB is used at ServiceNow, Google, Mozilla, and the Wikimedia Foundation.

– ngrok

A popular tool amongst web developers, ngrok allows the user to connect local servers hidden behind NATs and firewalls over secure tunnels to the public internet. Its main application involves remotely testing websites before final deployment by accessing them through a randomly generated link. In this project, ngrok has been used to display the application to the reviewers, gather data about its quality, the accuracy of the implemented emotional filter, and its usefulness.[19]

3.5. Tools necessary for model creation

– Jupyter Notebook

An open-source web application. It allows the user to create and share documents containing live code, equations, visualizations, and narrative text. As a result, Jupyter Notebook is frequently used in tasks like data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning and more. Jupyter Notebook is a product of Project Jupyter, an organization that exists to develop open-source software, open standards, and services for interactive computing across multiple programming languages.[20]

– Keras

A high-level neural networks API written in Python. Since its creation, the main idea behind Keras was enabling the user the fastest possible experimentation process. Aside from that, Keras supports easy prototyping, the implementation of both convolutional and recurrent networks and runs on CPU as well as a GPU.[21]

– TensorFlow

An end-to-end open-source machine learning platform. Its core feature is a library focused on the development and training of machine learning models. TensorFlow is used in projects by companies like Google, Intel and DeepMind.[22]

– pickle

Python's pickle module implements binary protocols used to serialize and de-serialize a Python object structure. In the case of this project, it's been employed as a way to use the previously trained BiLSTM model inside of the web application. "Pickling", as a process, refers to converting a Python object hierarchy into a byte stream. "Unpickling", on the other hand, is used to indicate the opposite operation: converting a byte stream back into object hierarchy.[23]

– dill

Dill extends pickle's module to allow the serialization and de-serialization of python objects for the majority of built-in python types. It provides the same interface as the pickle module, but also includes some additional features, such as the ability to save the state of an interpreter session in a single command. Dill is a part of the pathos framework.[24][25][26]

– GloVe

GloVe, or Global Vectors for Word Representation, is an unsupervised learning algorithm made to obtain vector representations for words. GloVe is trained on aggregated global word-word co-occurrence statistics from a corpus. In this project, a GloVe vector, pre-trained the Wikipedia 2014 and Gigaword 5 datasets, had been used. [27]

– Transformers

Known formerly as pytorch-transformers and pytorch-pretrained-bert, Transformers provide state-of-the-art general purpose-architectures (BERT, GPT-2, RoBERTa) for Natural Language Understanding (NLU) and Natural Language Generation (NLG) with over 32 pre-trained models for TensorFlow 2.0 and PyTorch. Transformers include a single AdamW optimizer, which matches PyTorch Adam optimizer API and allows the user to take advantage of standard PyTorch or apex methods for the schedule and clipping.[28][29]

– simple transformers

A library built as a wrapper around the above mentioned Transformers library. Currently supports Sequence Classification (binary, multi-class, multi-label), Token Classification (Named Entity Recognition) and Question Answering.[30]

4. Dataset choice and preparation

4.1. Dataset limitations and modeling criteria

Unfortunately, using more sophisticated models of emotions for this project (such as the circumplex model of affect) has proven to be impractical in the context of this project for several reasons:

1. Headline length and character

   News headlines tend to be both short and highly informative. Much of their emotional impact is not defined by simple expressions, such as “sad“ or “happy“, but by their cultural context. This tends to both increase the complexity of the problem and decrease the model’s effectiveness, and serves as a reason to avoid filters based on dictionaries,

2. Number of labels

   The maximal accuracy of the predictions tends to decrease with label count. In the situation of dataset scarcity and already high difficulty of the problem, the decision to further decrease possible usefulness of the model does not appear to be the right decision,

3. Dataset availability

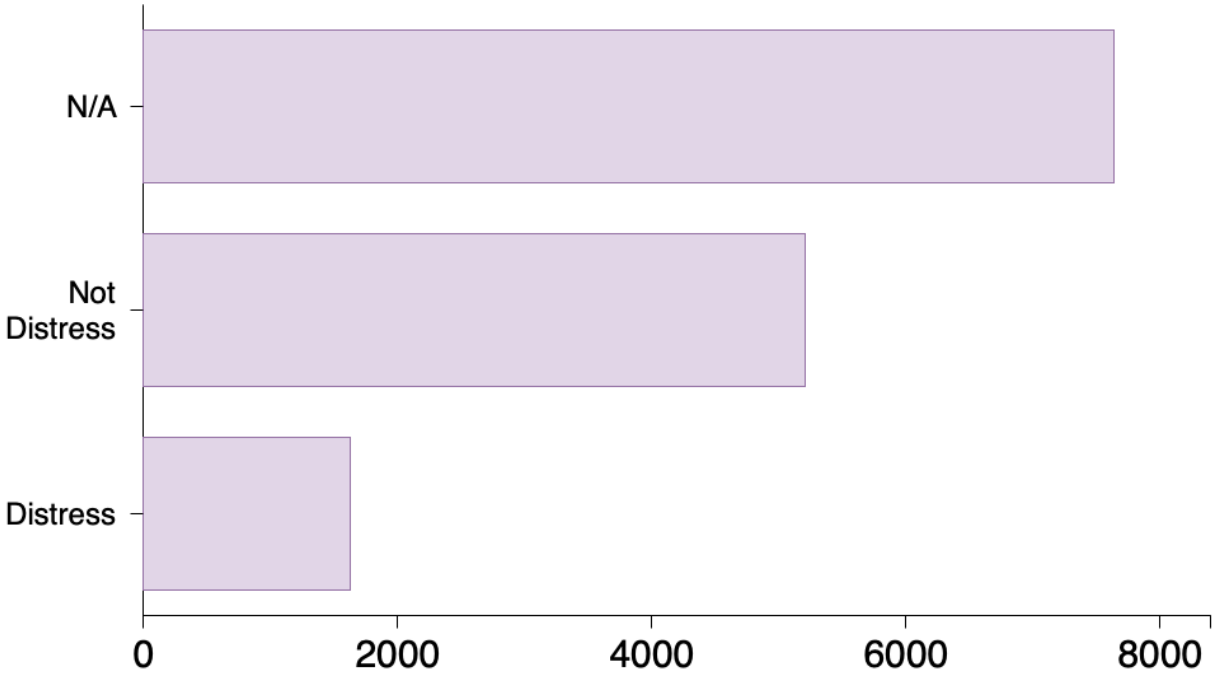
   Another constraint is defined by dataset availability. The aforementioned problems in news headline classification suggest the need for a highly specified dataset, both for testing and training. Such datasets tend to be harder to find for more complicated models of emotions and are very time- and resource-consuming to produce.

4.2. The problem of context

As much as it is important to acknowledge the problem of sensationalism in media, it is also crucial to understand that properly understanding and classifying news requires acknowledging the context of the information presented in it. That already proves to be a complicated task, since the nature of context in the news is both vast and ever-changing. This warrants the need for both a new and very varied dataset.

Using the Google data search tool, it has been determined that there are no large enough news headline datasets labeled with their sentiment, relevant to multiple categories. Most of the available datasets

Rys. 4.1. Examples from the dataset used in this project in the form of a JSON file



Rys. 4.2. Proportion of different labels in dataset

involve only news focused on the topic of finance, and therefore do not provide enough context to train a model capable of recognizing the majority of every-day news.

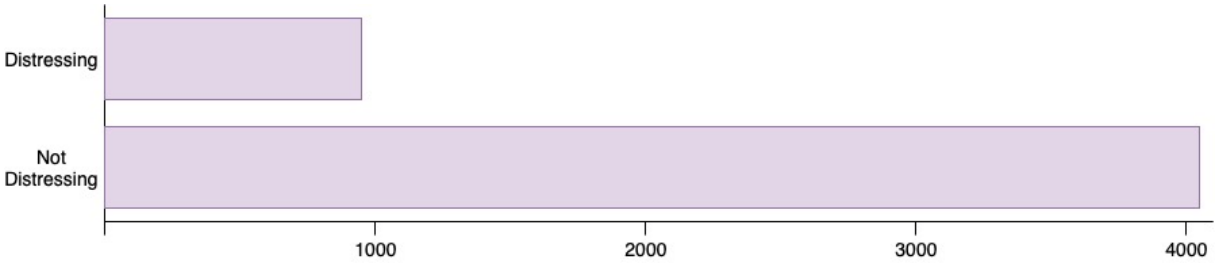
Therefore, two datasets have been used in this project. First dataset has been chosen as a basis for experimentation with two different neural network architectures and the creation of the first classifier, with focus limited only to one category of news. This dataset comes from Open Data Turks and includes 14555 examples of various financial news headlines, labeled as “Distressing“, “Not Distressing“ and “N/A“. The labels directly correspond to the emotional distress caused to the reader, instead of only revealing some form of correlation, like the “positive“/“negative“/“neutral“ choice.

The second dataset has been formed with the aid of the first classifier, by automatically labeling the data and residing to the manual labor only as a way of adjusting and correcting wrong examples in the dataset. The final, implemented classifier has been trained on this dataset.

It consists of 5000 sample headlines, chosen from the News Category Dataset, available on Kaggle. The dataset incorporates headlines from Huffington Post, published in the years 2012 to 2018. I have chosen the categories of politics, healthy living, wellness, entertainment, and parenting. All of the categories belong to the 10 most common in the dataset, and, by proxy, on Huffington Post.

4.2.1. Dataset preparation

Because the only available on DataTurks data format is a JSON file, the first dataset used in this project had to be converted to the format of pandas.Dataframe[31]. The second has been stored as a CSV. Unnecessary metadata has been deleted from both, which included information about the time of labeling the row as well as hashed messages added to the headline.



Rys. 4.3. Second, binary dataset

4.2.2. Text preprocessing

The simpletransformers already implements specific text preprocessing for BERT and RoBERTa models. As a result, before training these models only information about the line breaks had to be removed from the headlines. Dataframes applied to the BiLSTM model needed to first undergo the process of removal of tags, punctuation, single characters, and multiple spaces. Instances of unlabeled titles have been accounted for. The headlines have also been padded to achieve the same maximal length.

4.2.3. Label encoding

Because the first dataset used in this project, the FinNews Dataset from DataTurks, contains three categorical labels, two of which can be applied simultaneously to the same headline, applying a one-hot encoding proved to be necessary. Each set of labels in it is represented as a three-dimensional, sparse vector with values of zero for every column, except for the ones to which a label is applicable, which receives the value of one. The advantageous aspect of this implementation is the ease, with which a machine learning model can incorporate such categorical feature information by only learning one separate parameter for each dimension.[32]

Labels in the second dataset can only have two values: “Distressing“ or “Not Distressing“. Therefore, they would only need to be represented as a field with the value of either zero or one.

4.2.4. Tokenization

Known also as “the process of breaking a stream of textual content into words, terms, symbols, or some other meaningful elements called tokens“[33]. It is a technique widely used in linguistics as well as data science, where it generally occurs during text preprocessing at the word level. In this project, a Keras tokenizer is used to tokenize headlines.

4.2.5. Word embeddings

Known also as distributed word representations[34], word embeddings have found many applications in Natural Language Processing. Their main advantage over the more widely applied in machine learning one-hot encoding is the dimension reduction, much needed in the case of text representation. Conveying

the multitude of semantic and syntactic relations between words can be a computationally daunting task, and there are a few methods to make it achievable.

The one chosen in the case of this paper is GloVe, meaning Global Vectors[27]. Instead of basing its evaluation on the distance between pairs of word vectors, the semantic vector space of GloVe has been measured by examining the difference in many dimensions. As a result, GloVe produces a specific weighted least squares model, that achieved state-of-the-art 75% accuracy an outperformed other current methods on word similarity task.[27]

GloVe embeddings were used only in the creation of the BiLSTM model.

5. Model architecture and training

For this project, I have decided to experiment with two model architectures: a Bidirectional LSTM and a pretrained RoBERTa based transformer. Both are commonly used in sentiment analysis, with the second being a relatively new innovation (the transformer architecture has been proposed as a solution to Natural Language Processing tasks in 2017[28]).

5.0.1. Sequence transduction

A transducer converts a signal into another form. In statistical learning theory, transduction or transductive learning is used to describe predicting specific examples given previous examples from a knowledge domain, such as labeling the topic of an article based on a dataset of labeled articles. Therefore, the problem of labeling news articles based on their distressing effect can also be understood as solved by transduction.

5.0.2. Long-short Term Memory

Long-short Term Memory (LSTM) network architecture has been developed as a solution to the frequent problems encountered in Recurrent Neural Networks (RNN)[37], such as:

– Gradient exploding problem

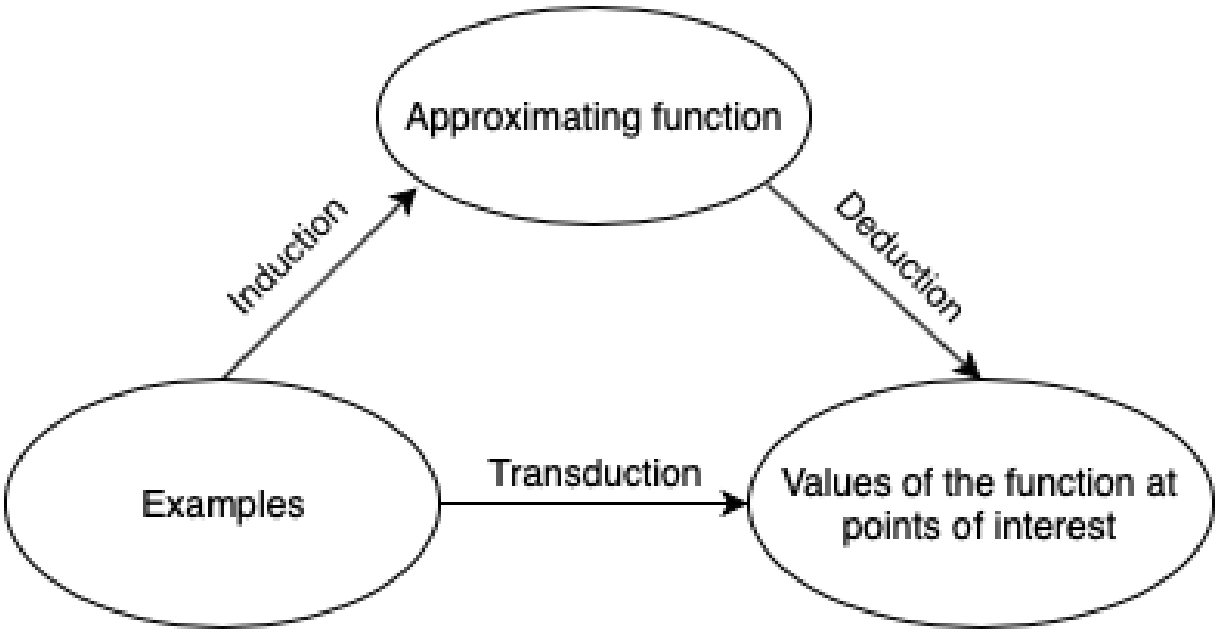
  It refers to the large increase in the norm of the gradient during training[38]. This kind of event is caused by the explosion of long term components, which have the ability to grow exponentially more than short term ones.

– Gradient vanishing problem

  Refers to an unwanted behavior opposite in nature, in which long term components' values drop exponentially fast, achieving norm 0 and making it impossible for the model to properly learn.[38]

LSTM can be understood as a type of RNN with gated structure, designed to learn long-term dependencies of sequence-based tasks[39]. They have been used in speech recognition[40], image captioning[41], music composition[42] and human trajectory prediction[43].

Basic RNN and LSTM architecture differ only in the incorporation of one crucial component: the hidden layer, also known as LSTM cell.



Rys. 5.1. Relationship between Deduction, Induction and Transduction. Illustration based on "The Nature of Statistical Learning Theory"[35] and "Machine Learning Mastery"[36]

The LSTM cell architecture consists of a cell memory state and three gates: an input gate, an output gate and a forget gate. The gated structure, with a special emphasis on forget gated, significantly improves LSTM's performance as a solution to several learning problems involving sequential data.[44] Each cell inside it can be computed, using a set of equations as follows[37]:

$$X = \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \tag{5.1}$$

$$f_t = \sigma(W_f \cdot X + b_f) \tag{5.2}$$

$$i_t = \sigma(W_i \cdot X + b_i) \tag{5.3}$$

$$o_t = \sigma(W_o \cdot X + b_o) \tag{5.4}$$

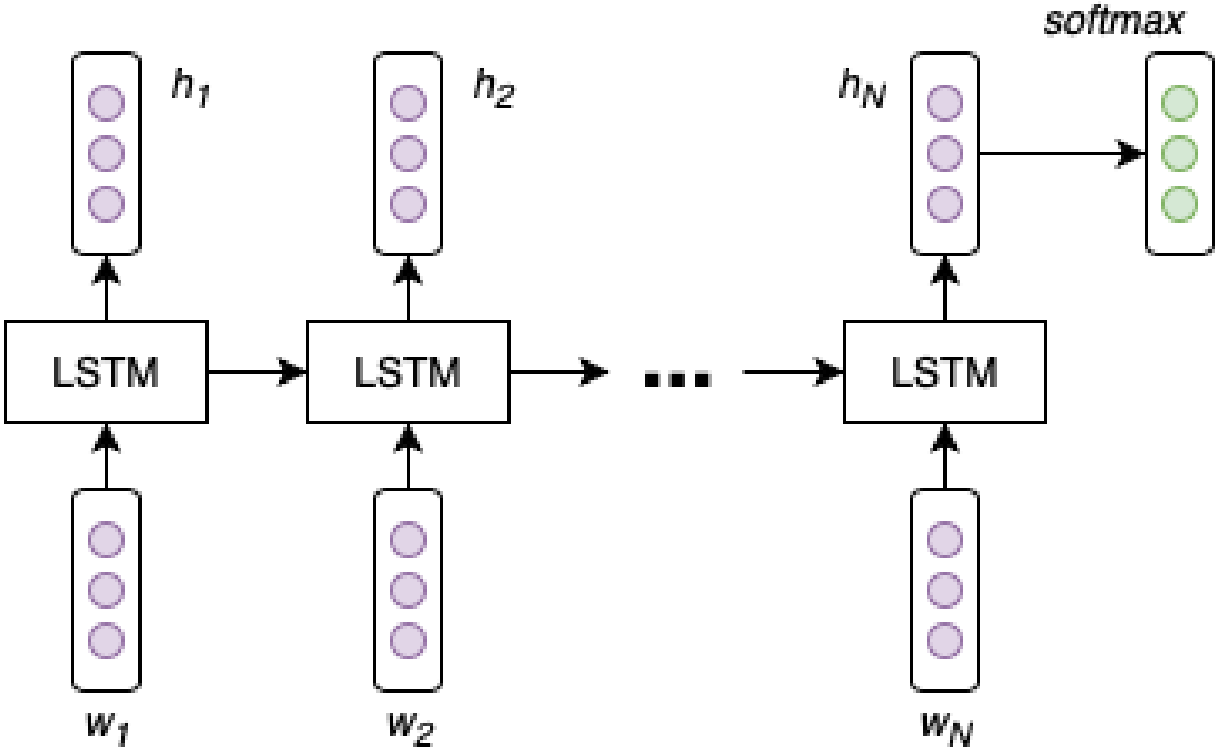
$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c \cdot X + b_c) \tag{5.5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{5.6}$$

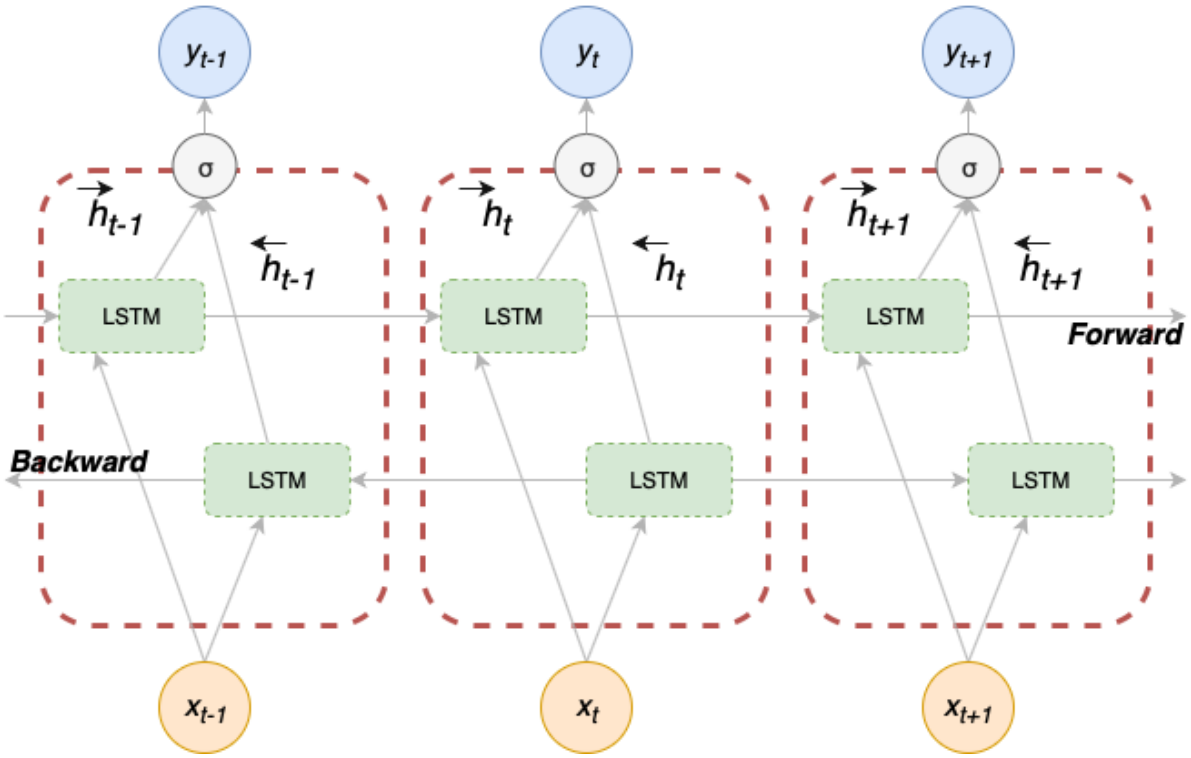
5.0.3. Bidirectional LSTM

A bidirectional LSTM (BiLSTM) has the ability to deal both with the backward and forward dependencies. This improvement is caused by a significant change in model architecture - connecting two

$W_i, W_f, W_o \in R^{d \times 2d}$ represent the weighted matrices and $b_i, b_f, b_o \in R^d$ the biases to be learned during training, the parametrization of the transformation of the input and the forget and output dates respectively. $\odot$ marks element-wise multiplication and σ the sigmoid function. multiplication. x_t consists of the inputs of the LSTM cell unit, representing the word embedding vectors w_t as shown in the illustration below. h_t is the vector of hidden layers. h_N, as the last hidden vector, is regarded as the representation of sentence and, after being linearized ad a vector of length equal to the number of class labels, serves as an input to the $softmax$ layer.



Rys. 5.2. An illustration of the architecture of a standard LSTM. A sentence of length N is represented as a word vector with weights $\{w_1, w_2, ..., w_N\}$. $\{h_1, h_2, ..., h_N\}$ describe the values of the hidden vector. Based on the description in "Attention-based LSTM for Aspect-level Sentiment Classification"[37]



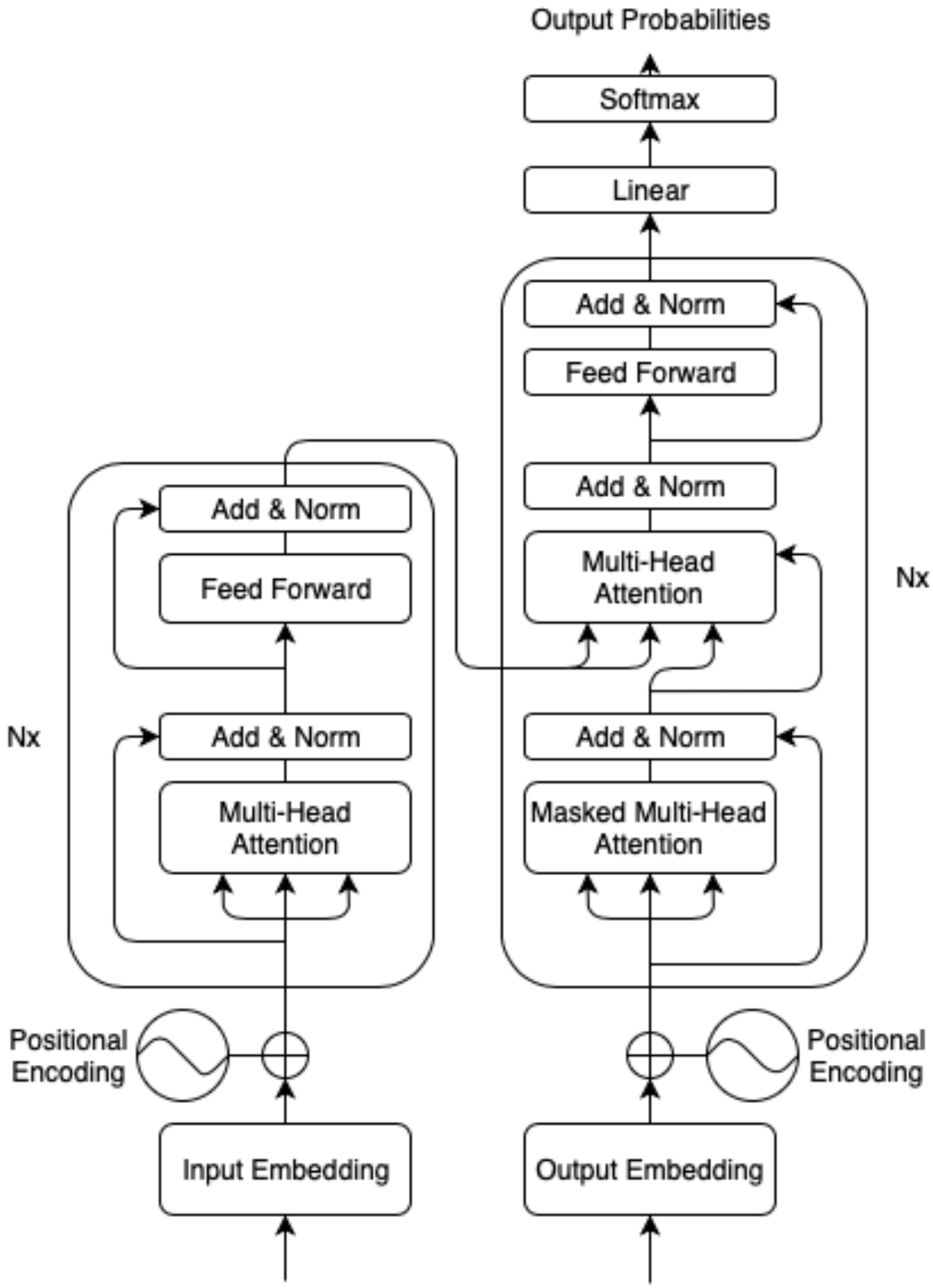
Rys. 5.3. Bidirectional LSTM architecture. Illustration based on "Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction"[39] [45]

hidden layers to the same output layer. The idea for this direction of LSTM development comes can be traced to bidirectional RNNs.

The illustration above can serve as a description of an unfolded BiLSTM layer, containing a forward and backward LSTM layer[39]. The forward layer output sequence, $\overrightarrow{h}$, is calculated iteratively, using inputs in a positive sequence from time $T - n$ to time $T - 1$. Meanwhile, the backward layer output sequence, $\overleftarrow{h}$, can be computed using the reversed inputs from time $T - n$ to $T - 1$. Both layer outputs are obtained by using standard LSTM equations, shown as Equations (1) - (7).

5.0.4. Transformers

Contrary to Convolutional Neural Networks, examples of which were described above, Transformers do not rely on complicated architecture in order to capture the sequential nature of their input information. Instead, they employ an attention mechanism, knows also as self-attention. By doing so, they enable the system to avoid the fatal flaw of RNN- based approach - the inability to parallelize the sequentially-computed network. The main consequence of this problem is a high difficulty of implementation for long sequences, and, as a result, a constraint on the batch size that can be used during training. [45]Removing this issue has recently allowed the development of giant, powerful models like BERT and RoBERTa.



Rys. 5.4. Basic transformer architecture. Illustration based on [45]

A transformer's architecture contains the Encoder-Decoder framework, encountered in earlier Attention networks. It accepts the input sequence, transforms it into an encoding based on the context and decodes the encoding in the output. Instead of LSTMs, a transformer includes a Self Attention layer and a tool used to identify the sequential nature of the data: Positional Encoding. The fact, that all components of a transformer are made of fully connected layers, allows for easy parallelization and so, wider distribution.

5.0.5. BERT

BERT (or Bidirectional Encoder Representations from Transformers) is a relatively new language representation model, designed to pre-train deep bidirectional representations from the unlabeled text by jointly conditioning on both left and right context in all layers. It is possible to use a pre-trained BERT model with just one additional output layer added for fine-tuning as a state-of-the-art model for a variety of tasks, including multi-label classification.

BERT improves and expands on the basic Transformer model by introducing bidirectionality, using a "masked language model" (MLM) pre-training objective. The masked language model masks some of the input tokens at random and introduces the objective of correctly predicting the original vocabulary id of the masked word based on its context.

BERT uses WordPiece embeddings with a 30000 token vocabulary. For the pre-training corpus, it utilizes the BooksCorpus and English Wikipedia corpus (though while extracting only the text passages and ignoring lists, tables, and headers).

Contrary to the previous direction of Natural Language Processing's development, model size seems to warrant much more effective results than a sophisticated model architecture even on very small scale tasks (providing the model has been sufficiently pre-trained).[46]

5.0.6. RoBERTa

RoBERTa (also known as A Robustly Optimized BERT Pretraining Approach) is a retrained BERT model with significantly higher performance according to metrics like GLUE, RACE, and SQuAD. In comparison to BERT, RoBERTa had been trained longer, with bigger batches, with more data, and on longer sequences. The next sequence prediction objective from BERT has been entirely removed. Moreover, the masking pattern had been dynamically changing, applied to train data. In training, the following corpora have been used:

– BookCorpus plus English Wikipedia

  The original data used to train BERT.

– CC-News

  Collected from the English portion of the CommonCrawl News dataset. CC-News contains 63 million English news articles crawled between September 2016 and February 2019.

– OpenWebText

  A dataset containing web content extracted from URLs shared on Reddit with three upvotes and more. An open-source recreation of the WebText corpus.

– Stories:

  A dataset containing a subset of CommonCrawl data, filtered to match the style of Winograd schemas.

RoBERTa used a larger byte-level BPE (Byte-Pair Encoding) vocabulary, containing 50K instead of 30K subword units, without any additional preprocessing or tokenization of input.[47]

5.1. Model training

To summarize, during the development of this project, three different models have been evaluated and trained.

The first one is a Bidirectional LSTM implemented in Keras. Training of the model has been conducted on 80% of the dataset, with 20% of it left for evaluation, for 5 epochs (with epoch being a term, indicating the number of passes through the entire training dataset the algorithm will complete) and with the batch size of 128 (batch size meaning the number of training examples, used in one iteration of the algorithm). The model, as will be further described below, has performed relatively well on the first, limited to finances dataset. However, the burdensome serialization of the model, the need for elaborate text preprocessing and the lack of the self-attention mechanism have motivated further experiments with the transformer architecture.

The second model, trained on the same dataset, but with a pre-trained, RoBERTa-based architecture for multi-label classification from simpletransformers[30], achieved a larger accuracy while training for only 1 epoch. The accuracy and serialization of the model seemed acceptable, but the lack of variety of data in the dataset it has been trained on prevented it from exhibiting an effect on the average newsreader. The fact, that everyday news belong mostly to categories like politics, healthcare or entertainment means, that the financial classifier would only make an impression on especially avid business news consumers.

The third model, trained on the new, specifically created for this purpose dataset, has been trained on 3 epochs. Its architecture is also RoBERTa-based, as was its predecessor's, but the problem it solves is an example of binary classification, not multi-label (the "N/A", as a label meaning not applicable to financial distress classification, has not been taken into consideration during the creation of the second dataset). The model achieved decent performance for its small training dataset, both according to metrics and opinions of readers, expressed later in the questionnaire.

6. Aggregator design

Among the main priorities set for the aggregator design have been simplicity, user-friendliness, and portability. Because its function is mostly to showcase the abilities of the model, the features it implements should be few and well-thought-out. The most general and important is decreasing the stress levels of the user, which suggests a minimalist design and focus on easy navigation. The vast possible number of different models, classifiers and filtering methods indicates the need for the portability of the distress level measurement.

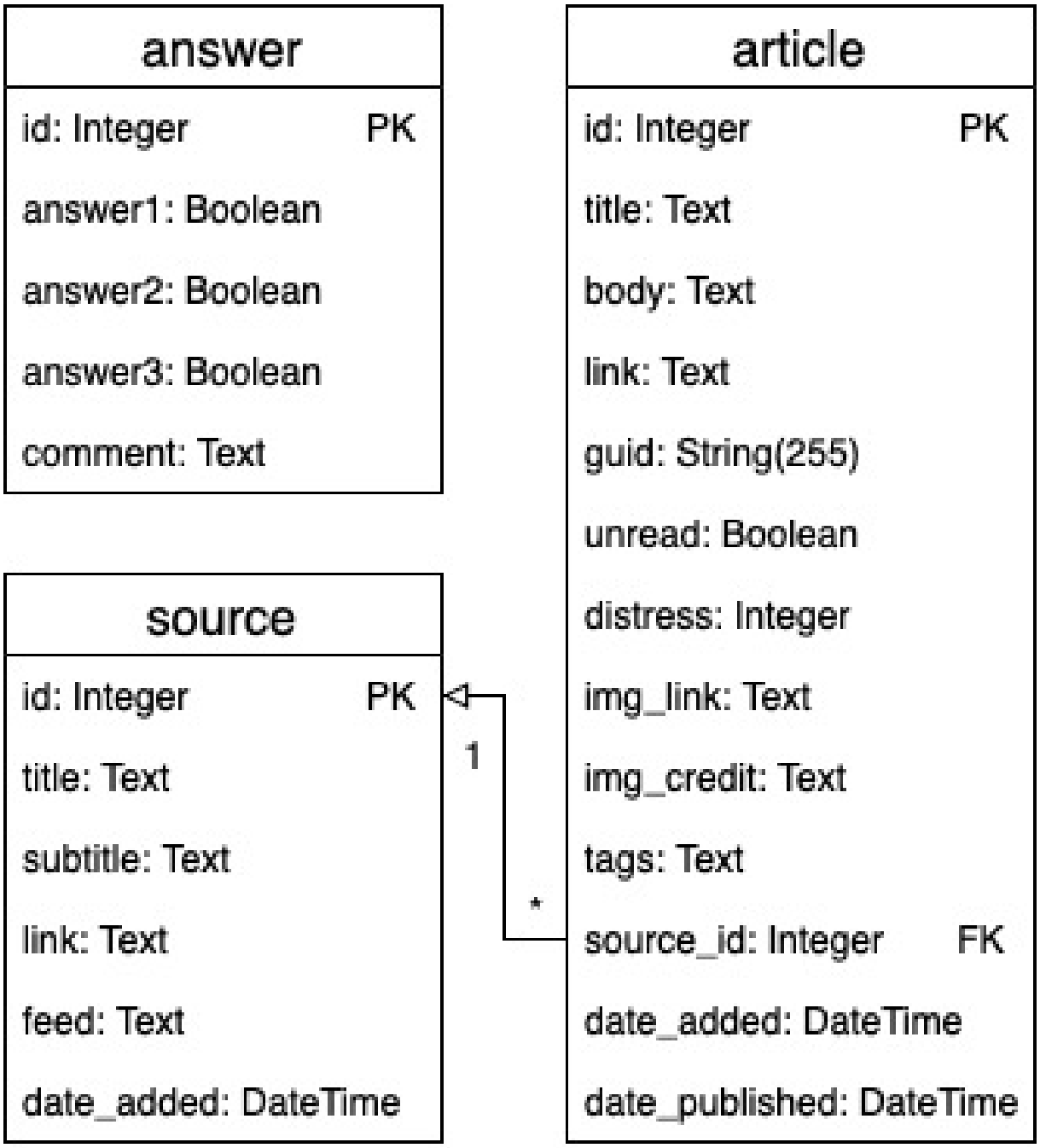
6.1. Database architecture

As seen on the illustration, the database proposed for this project can be seen as relatively simple and small. It consists only of three tables: the article, containing all the articles from all of the subscribed feeds, source - a separate table for all of the sources of the articles and answer - a disconnected table created only to gather information from the questionnaires.

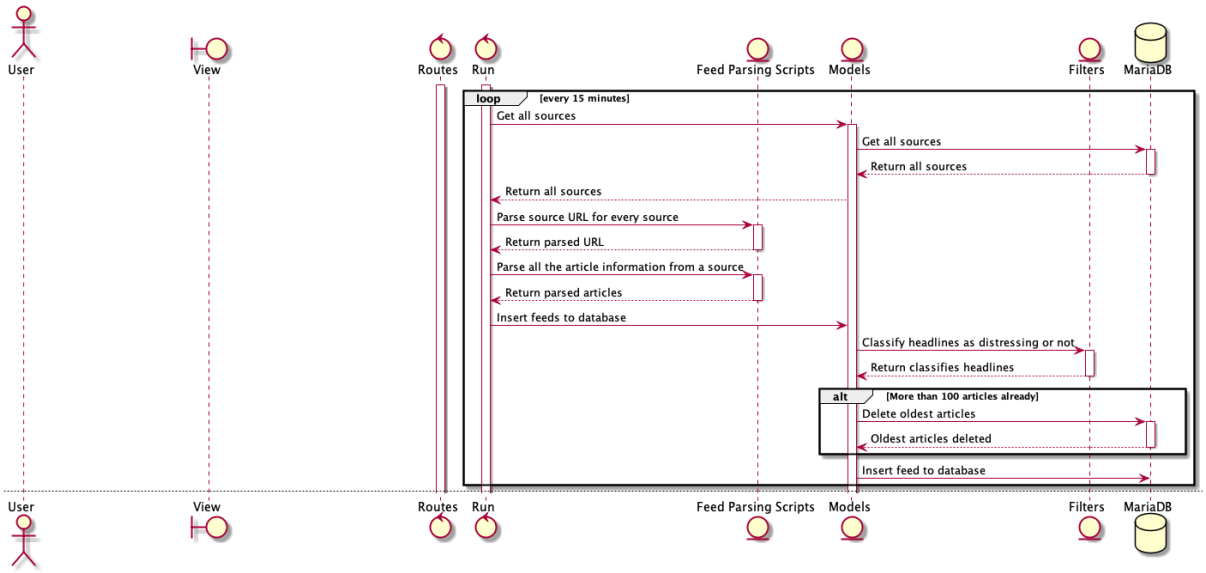
6.2. Basic features

Among the basic features necessary to test the aggregator, unrelated to any personalization or modification of the article list by the user, is the systematic updating of the article list. As a news aggregator, it cannot only gather an old, unchanging collection of articles, but should systematically add new positions to the list to keep the feed exciting and relevant.

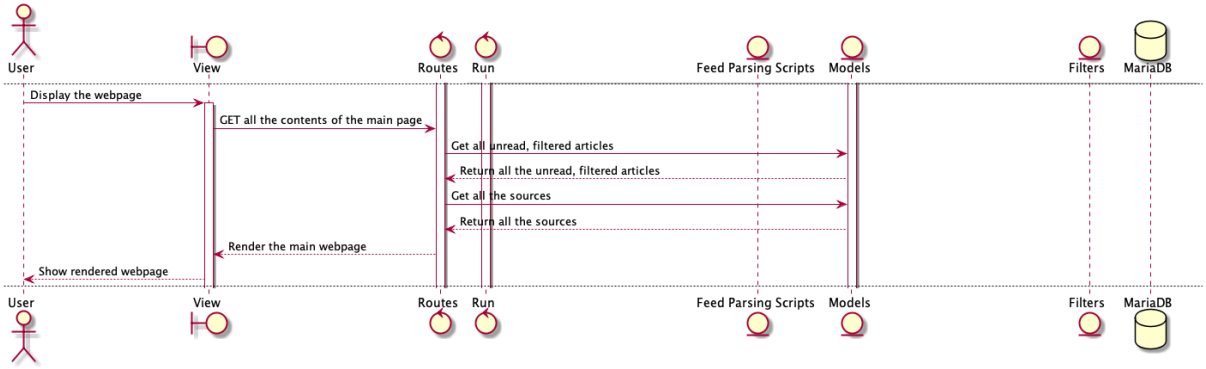
```
def updating_loop():
    while True:
        with app.app_context():
            q = source.Source.query
            for src in q.all():
                try:
                    update_source(src)
                except:
```



Rys. 6.1. Database used in the project



Rys. 6.2. The sequential diagram of the updating process



Rys. 6.3. The sequential diagram of displaying the main page

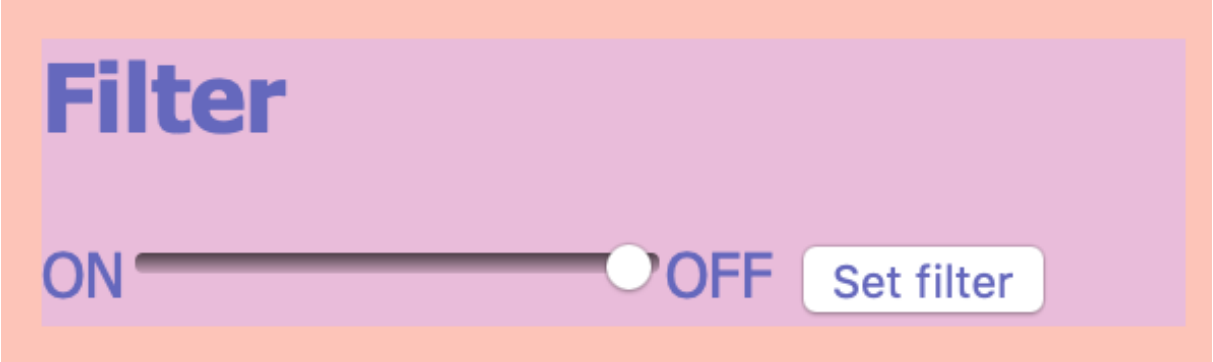
```
                continue
    time.sleep(15*60)
```

The main loop of the application contains a simple updating loop, whose purpose is to update the article list by adding recent articles from provided sources. The regular update repeats every 15 minutes, with the intent of appearing often enough to keep the news list engaging without having to implement continuous updates.

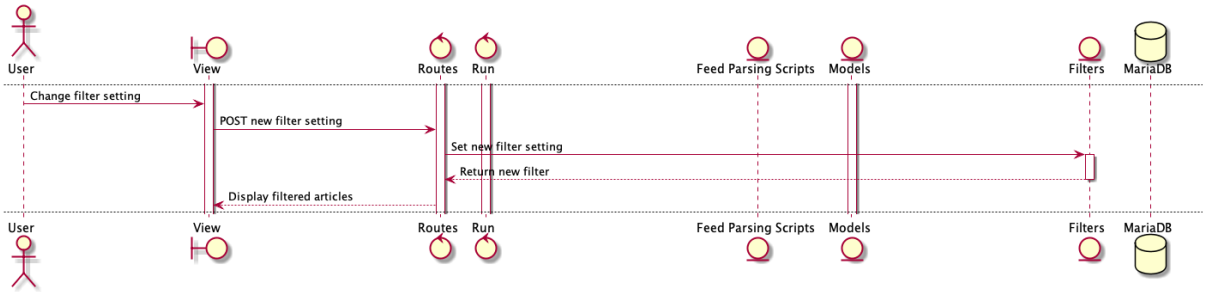
Another basic feature is connected with the simple activity of displaying the main page, with the small, additional requirements of showing only the unread, filtered articles.

6.3. Additional features

An explicit toggle allows the user to explore both filtered and unfiltered articles from the feeds, depending on their mood and momentary preferences. The filter is meant to be used primarily for especially



Rys. 6.4. Filter



Rys. 6.5. The sequential diagram of the filtering change

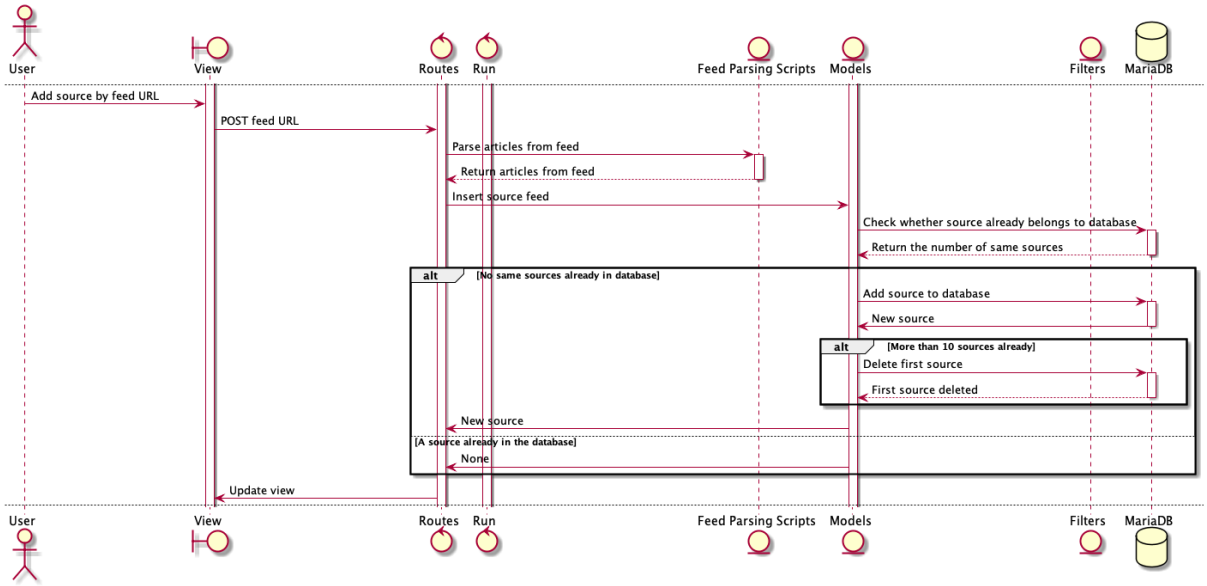
stressful situations and times of the day. In other cases, users can choose to browse through the whole news stream in its original form, and switch back as soon as they judge themselves to be not ready for a non-filtered message stream.

```
def source_get(parsed):
    ready_feed = parsed['feed']
    return {
    'link': ready_feed['link'],
    'title': ready_feed['title'],
    'subtitle': ready_feed['subtitle'],
    }
```

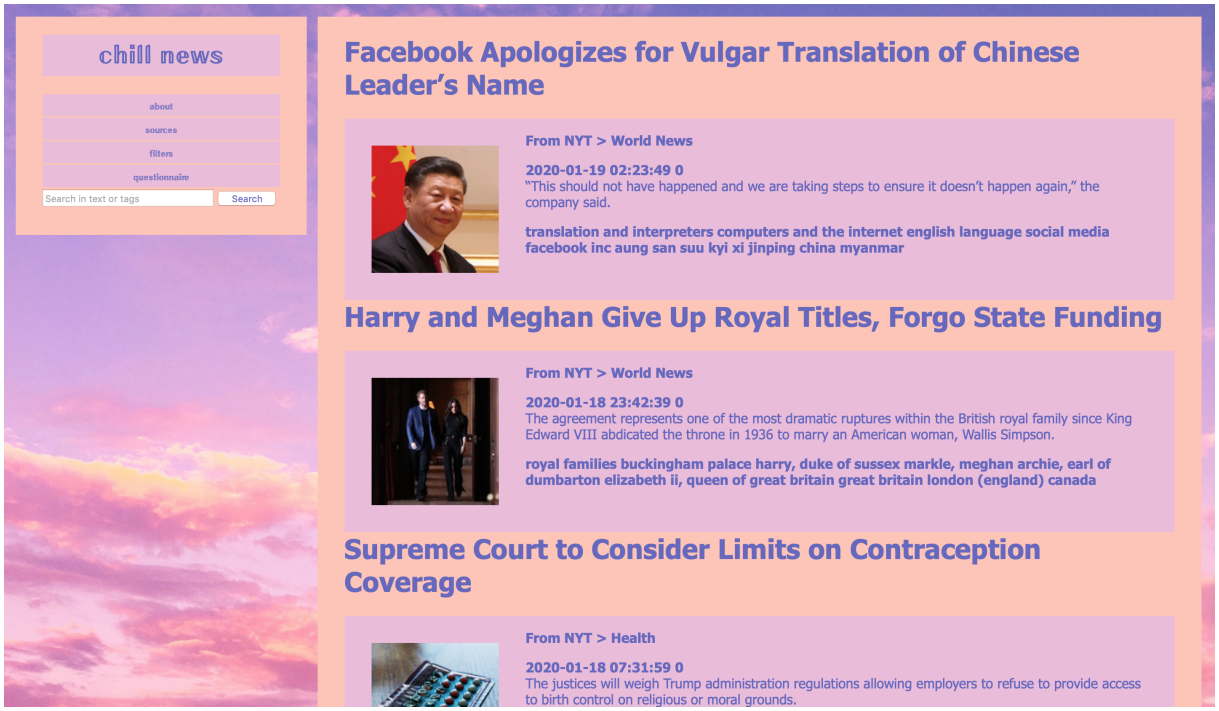
Another small feature added to the aggregator is the user's ability to see all the sources and expand their list by providing a suitable RSS feed URL to the form. The user cannot increase the sources list to more than 20 elements (in the event of attempting to add the 21st source, aggregator deletes the source with the earliest date_added). It is also impossible to add a source twice.

The visual design of the aggregator has been created with the implicit purpose to be both soothing and easily recognizable. The choice of pinks and pastels was meant as a tribute to the vaporwave style of early Internet culture. However, due to negative feedback from the readers, new versions will include a more natural and soothing color palette.

Rys. 6.6. Sources



Rys. 6.7. The sequential diagram, depicting the addition of the source



Rys. 6.8. View

7. Project evaluation

7.1. Metrics used in Multi-Label Classification

7.1.1. Label Ranking Average Precision Score

The main metric used in the project to measure the precision of models, used to solve our multi-label classification problem, has been the Label Ranking Average Precision Score, known also as LRAP. LRAP is the average over each ground truth label assigned to each sample, which is equivalent to the ratio of correctly classified labels vs. all labels with a lower score.

The goal of this metric is to give a better rank to the labels associated with each sample. It's a good choice for our project because of the general characteristics of the problem (multi-label classification), but also as a way to mitigate the low amount of "Distressed" examples compared to "N/A".[48]

The LRAP score for the second model amounts to 86%.

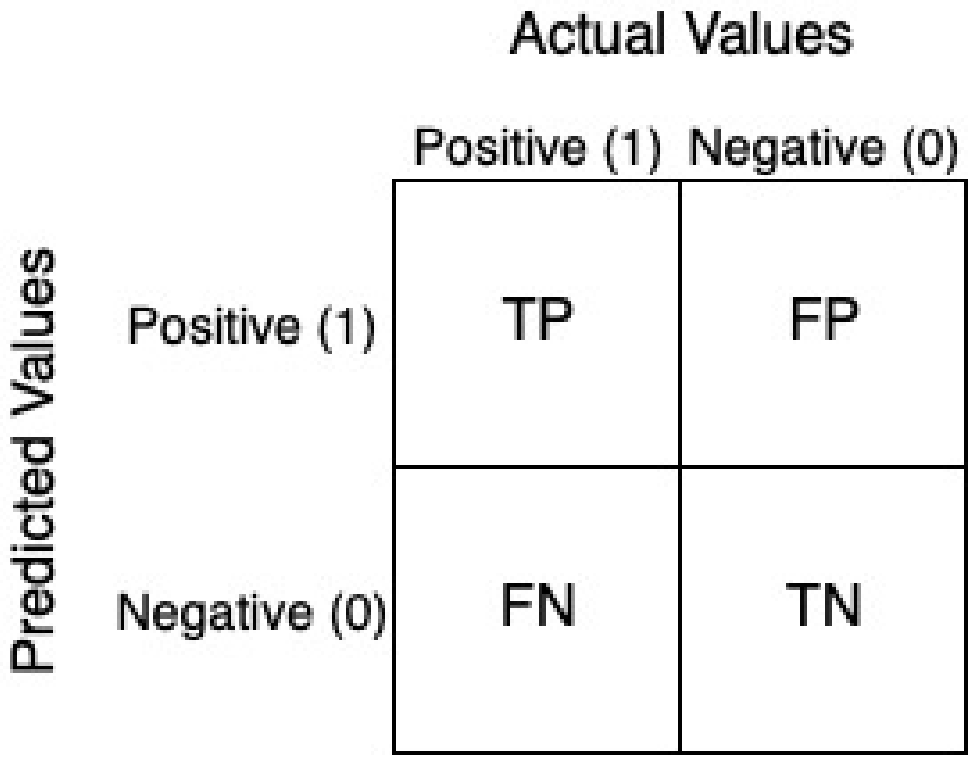
7.1.2. Zero Rule Algorithm

Commonly used as a benchmarking algorithm, Zero Rule finds the most commonly encountered class and classifies the input as such. It's an especially useful metric in case of unbalanced datasets, which cause many contextless algorithms to achieve high accuracy and fail at deployment. It allows to check whether the model even acknowledges the less encountered class.

```
import numpy as np
from sklearn.metrics import label_ranking_average_precision_score

def zero_rule_algorithm(train_df, eval_df):
    outputs = [t for t in train_df['labels']]
    most_common = max(set(outputs), key = outputs.count)
    return [most_common for i in range(len(eval_df))]

predictions = zero_rule_algorithm(train_df, eval_df)
values = [i for i in eval_df['labels']]
```



Rys. 7.1. Coefficient matrix

```
label_ranking_average_precision_score(values, predictions)
```

In the sample code shown above the Zero Rule Algorithm “trains“ on the training dataset by choosing the most common label in it (in this case, “N/A“). Then, the algorithm classifies every input as the most common label, which in the case of a strongly imbalanced dataset may be mistaken as just a result of a highly effective model,

The Label Ranking Average Precision Score for the Zero Rule Algorithm consequently places its value around 68.09%. The second model, based on RoBERTa, has achieved average precision higher by more than 10%. Therefore, we can conclude that the RoBERTa-based model does form predictions about the labels without exploiting the imbalance of the dataset.

7.2. Metrics used in Binary Classification

7.2.1. Matthews correlation coefficient

The Matthews correlation coefficient is a metric used in machine learning, proposed by Brian W.Matthews as a way to better represent the quality of binary classification.[49] It is recognized as a more revealing metric than accuracy because it incorporates the balance ratio of all four confusion matrix categories: True Positive, True Negative, False Positive and False Negative.[50]

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7.1}$$

As can be noticed in the equation above, accuracy as a metric does not fully consider the size of the classes in computation, focusing rather on sums of correct and faulty predictions. Because of that, it fails when applied to problems with huge differences in the size of the classes. A model with no True or False Positives can achieve high accuracy by always predicting the negative, just because of the negative outcome being so much more common than the positive one. Of course, this behavior can be very unhelpful in problems like cancer detection or some emotional filters.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{7.2}$$

Matthews correlation coefficient, on the other end, varies from -1 to 1 - with 0 determining a completely unhelpful, not correlated to the data model. The binary simpletransformers model achieved 0.49 in Matthews correlation coefficient, with 108 True Positives, 739 True Negatives, 74 False Positives, and 79 False Negatives.

7.2.1.1. Questionnaire

As a natural consequence of the psychological nature of the project, in order to complete the evaluation, a subjective measure of the aggregator's usefulness had to be introduced.

To reduce the variance of possible answers, the subjects of the first three inquiries have been reduced to the bare essentials of this project:

– The efficiency of the filter

  The purpose of the question "Has the aggregator managed to filter out negative headlines?"is to confirm, that the efficiency of the classifier, measured according to previously established metrics, corresponds to its subjective efficiency for the average reader.

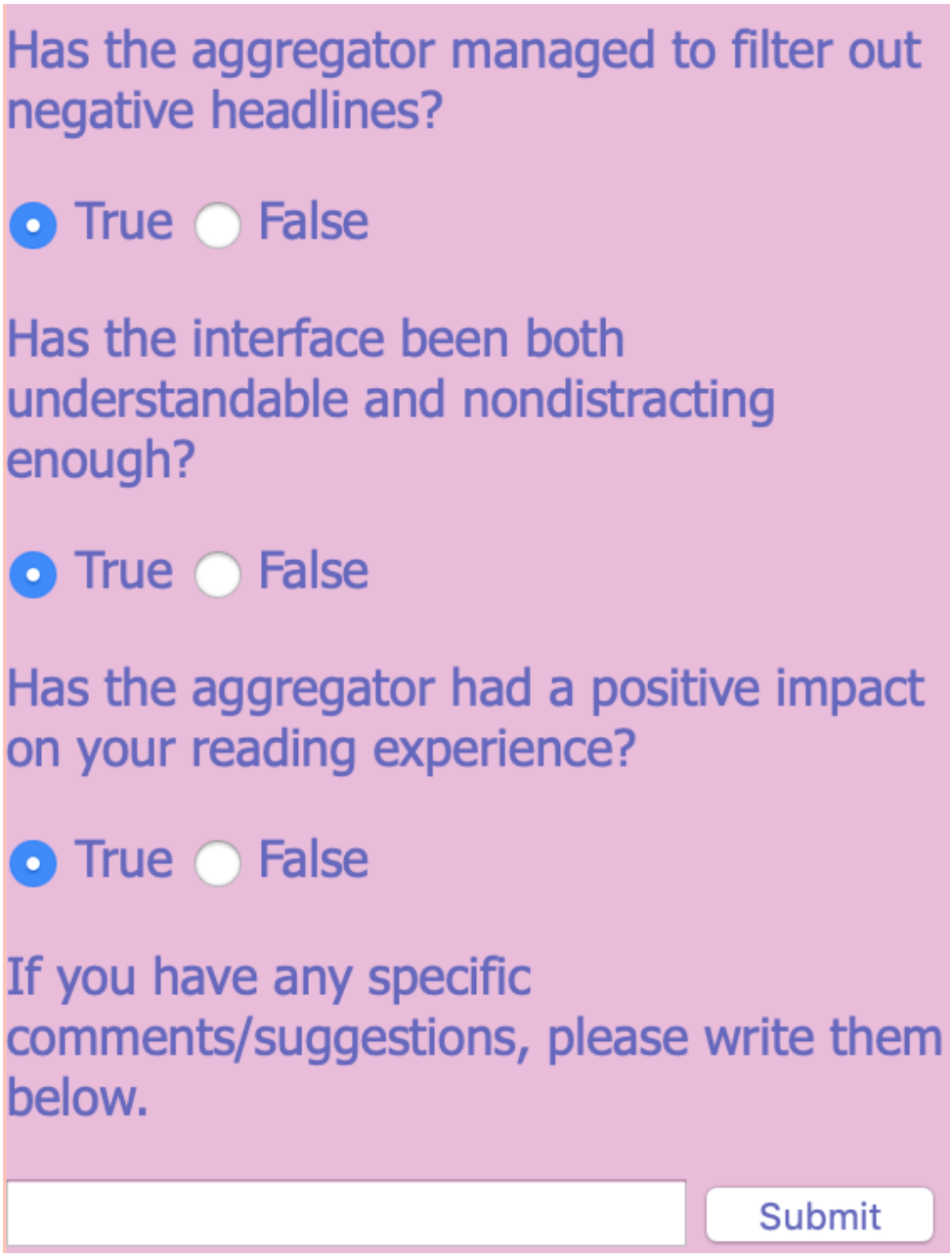
– The usability of the aggregator

  The question "Has the interface been both understandable and nondistracting enough,, allows the user to express, whether they consider the aggregator's implementation and design as sufficient for its purpose.

– The positive (or not) impact of the project

  The last question, "Has the aggregator had a positive impact on your reading experience?", strives to confirm, that not only the components of the project seem to be made with sufficient forethought and precision, but the whole product can be considered as yielding beneficial results.

To allow the readers to fully express their personal opinion and offer their suggestions, a comment section has also been added to the form.



Rys. 7.2. Questionnaire

After allowing 8 people both the time and access to test the application, all of them have formulated similar answers. In every case, all three questions have been marked as true. However, the comment section has gathered a considerable amount of suggestions, most of which involving the website's visual design, such as "I propose a more neutral color palette", "no images for some articles" and "indentation is too large in the paragraph". Others reference the website's functioning, suggesting the addition of a feature, allowing the reader to exclude specific entities or words and as a result filter out their triggers.

Both groups have valuable input, especially so in the case of expanding the application. However, the results of the first part of the questionnaire suggest, that the minimum viable version of the product has already been achieved, enough to ensure a more relaxing news reading experience for the user. Other, not as necessary modifications will be added in the next versions.

8. Conclusion

8.1. Achieved goals

Basing the judgment on the result of the questionnaire and various metrics, already described in the previous chapter, the goal of creating a minimal viable filtering application has been achieved. The results of the evaluation also show a possible demand for the product of the news filtering aggregator.

8.2. Further development

Aside from changes, mentioned previously in the questionnaire section, there are a few directions the further development of the project could follow.

8.2.1. Modeling more complicated emotions

More sophisticated filtering models could be developed and used in the project. However, due to the need for highly specified data (correctly labeled news headlines) and the lack of both monetary and temporary resources, a simpler, more surface-level model has been used. Rich enough datasets could support incorporating more realistic models of emotions, which instead of simply discerning the distressing articles from the not distressing ones, could detect proactive or depressive emotions and act accordingly.

8.2.2. Developing filtering methods suitable for different languages

For now, the filtering methods used in this news aggregator allow the user to filter through the stream of English articles. A natural direction of development would involve enabling foreign language support, which would improve not only the comfort of the user but also the variety of points of view presented in the article stream.

8.2.3. Allowing the user to label news headlines

The classifier has been trained on a relatively small dataset, labeled according to the opinion of just a single person. News topics and headlines, however, tend to be changing rather quickly. Suddenly, new actors appear, referencing recent, controversial events, the sentiment of which is unfortunately unknown

to the model. One possible solution to this problem is gathering data from the users by providing a suitable button under every news abstract. That would both provide a more varied balanced dataset, labeled according to the opinion of many diverse people, but also keep the model relatively useful and accurate.

8.2.4. More options for self-customization of the aggregator

Another direction of further development, albeit not as focused on emotional filtering as the others, would be to provide more options for self-customization for the user. For example, adding more varied stylesheets, a list of sources specific to the user, separate accounts and lists of recently read and left to read articles. Albeit not crucial, those changes would probably improve the user experience and therefore indirectly reduce the stress involved with using the aggregator.

Bibliografia

[1] „February 22 to March 4, 2018 Pew Research Center Survey". W: ().

[2] Nic Newman. „Overview and Key Findings of the 2018 Report". W: ().

[3] „Compassion fatigue: coping with secondary traumatic stress disorder in those who treat the traumatized". W: ().

[4] „Is compassion fatigue inevitable in an age of 24-hour news?" W: ().

[5] *Compassion Fatigue: How the Media Sell Disease, Famine, War and Death*.

[6] *Against Empathy*.

[7] Liu Lin, Fan Xiaozhong i Zhao Xunping. „Research on Web monitoring system based on natural language processing". W: *International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003*. 2003, s. 746–751. DOI: *10.1109/NLPKE.2003.1276005*.

[8] Y. Peng i T. Chou. „Automatic Color Palette Design Using Color Image and Sentiment Analysis". W: *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*. 2019, s. 389–392. DOI: *10.1109/ICCCBDA.2019.8725717*.

[9] Magda Piórkowska i Monika Wrobel. *Basic Emotions*. Lip. 2017. DOI: *10.1007/978-3-319-28099-8_495-1*.

[10] „The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology". W: ().

[11] Yanghui Rao i in. „Building emotional dictionary for sentiment analysis of online news". W: *World Wide Web* 17 (lip. 2014). DOI: *10.1007/s11280-013-0221-9*.

[12] Lu Ye, Rui-Feng Xu i Jun Xu. „Emotion prediction of news articles from reader's perspective based on multi-label classification". W: 5 (2012), s. 2019–2024. DOI: *10.1109/ICMLC.2012.6359686*.

[13] Nicole Novielli Fabio Calefato Filippo Lanubile. „EmoTxt: A Toolkit for Emotion Recognition from Text". W: ().

[14] S. M. Al Masum, M. T. Islam i M. Ishizuka. „ASNA: An Intelligent Agent for Retrieving and Classifying News on the Basis of Emotion-Affinity". W: *2006 International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06)*. 2006, s. 133–133. DOI: *10.1109/CIMCA.2006.51*.

[15] „Mining and visualising information from RSS feeds: A case study". W: ().

[16] *Newspy Instructions (available inside the application)*.

[17] A.M. Kuchling. „PEP 206- Python Advanced Library". W: ().

[18] *MariaDB Server documentation*.

[19] *Ngrok documentation*.

[20] *Project Jupyter Documentation*.

[21] *Keras Documentation*.

[22] *Tensorflow Documentation*.

[23] *Python Documentation*.

[24] *Dill Project Documentation*.

[25] „"Building a framework for predictive science"". W: (2011).

[26] „"pathos: a framework for heterogeneous computing"". W: (2010-).

[27] Jeffrey Pennington, Richard Socher i Christopher D. Manning. „GloVe: Global Vectors for Word Representation". W: (2014), s. 1532–1543.

[28] *Transformers Library*.

[29] Thomas Wolf i in. „HuggingFace's Transformers: State-of-the-art Natural Language Processing". W: *ArXiv* abs/1910.03771 (2019).

[30] *simpletransformers library*.

[31] Wes McKinney. „Data Structures for Statistical Computing in Python". W: *Proceedings of the 9th Python in Science Conference*. Red. Stéfan van der Walt i Jarrod Millman. 2010, s. 51 –56.

[32] C. Seger. „An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing". W: (2018).

[33] „TEXT MINING: OPEN SOURCE TOKENIZATION TOOLS – AN ANALYSIS". W: *Advanced Computational Intelligence: An International Journal (ACII), Vol.3, No.1, January 2016* ().

[34] „Word Embedding Revisited: A New Representation Learning and Explicit Matrix Factorization Perspective". W: ().

[35] Vladimir N.Vapnik. „Statistical Learning Theory". W: ().

[36] W: ().

[37] Yequan Wang i in. „Attention-based LSTM for Aspect-level Sentiment Classification". W: sty. 2016, s. 606–615. DOI: *10.18653/v1/D16-1058*.

[38] Razvan Pascanu, Tomas Mikolov i Yoshua Bengio. „Understanding the exploding gradient problem". W: *CoRR* abs/1211.5063 (2012). arXiv: *1211.5063*.

[39] Zhiyong Cui, Ruimin Ke i Yinhai Wang. „Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction". W: *CoRR* abs/1801.02143 (2018). arXiv: *1801.02143*.

[40] „Speech recognition with deep recurrent neural networks". W: ().

[41] „Show and tell: A neural image caption generator"in Proceedings of the IEEE conference on computer vision and pattern recognition". W: (2015).

[42] „A first look at music composition using lstm recurrent neural networks". W: (2002).

[43] „Social lstm: Human trajectory prediction in crowded spaces". W: (2016).

[44] K. Greff i in. „LSTM: A Search Space Odyssey". W: *IEEE Transactions on Neural Networks and Learning Systems* 28.10 (2017), s. 2222–2232. ISSN: 2162-2388. DOI: *10.1109/TNNLS.2016.2582924*.

[45] Noam Shazeer Niki Parmar Jakob Uszkoreit Aidan N. Gomez Łukasz Kaiser Illia Polosukhin Ashish Vaswani Llion Jones. „Attention Is All You Need". W: ().

[46] Jacob Devlin i in. „BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". W: *NAACL-HLT*. 2019.

[47] Yinhan Liu i in. „RoBERTa: A Robustly Optimized BERT Pretraining Approach". W: *ArXiv* abs/1907.11692 (2019).

[48] *Scikit Learn Manual*.

[49] S. Boughorbel, Fethi Jarray i Mohammed El-Anbari. „Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric". W: *PLOS ONE* 12 (czer. 2017), e0177678. DOI: *10.1371/journal.pone.0177678*.

[50] Davide Chicco. „Ten quick tips for machine learning in computational biology"". W: *BioData Mining* (grud. 2017).