BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

NON-SUPERVISED SENTIMENT ANALYSIS

ANALÝZA SENTIMENTU BEZ PŘÍMÉHO UČENÍ S UČITELEM

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR — **JOZEF KARABELLY**
AUTOR PRÁCE

SUPERVISOR — **Ing. MARTIN FAJČÍK**
VEDOUCÍ PRÁCE

BRNO 2020

Brno University of Technology
Faculty of Information Technology

Department of Computer Graphics and Multimedia (DCGM) Academic year 2019/2020

Bachelor's Thesis Specification

Student: **Karabelly Jozef**
Programme: Information Technology
Title: **Non-Supervised Sentiment Analysis**
Category: Speech and Natural Language Processing

Assignment:

1. Describe the phenomenon of sentiment in natural language and identify its various problem perspectives as can be found in the current literature.
2. Research the current state-of-the-art of non-supervised methods (methods that do not use direct supervision) for given problem.
3. Describe available datasets for the problem.
4. Choose and describe a suitable method, describe how it differs from other related work.
5. Implement the proposed method.
6. Evaluate the method.
7. Do an ablation study.
8. Compare achieved results with state-of-the-art.

Recommended literature:

- Zeng, Z., Zhou, W., Liu, X. and Song, Y., 2019. A Variational Approach to Weakly Supervised Document-Level Multi-Aspect Sentiment Classification. *arXiv preprint arXiv:1904.05055*.

Requirements for the first semester:

- Complete items 1 to 4 of the assignment.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor: **Fajčík Martin, Ing.**
Head of Department: Černocký Jan, doc. Dr. Ing.
Beginning of work: November 1, 2019
Submission deadline: May 28, 2020
Approval date: November 1, 2019

Abstract

The goal of this thesis is to present an overview of the current state of research in the non-supervised sentiment analysis and identify potential research paths. Besides, the thesis introduces a novel self-supervised pre-training objective. Extending the model trained with the introduced objective with one extra layer of neural network and training it alone shows promising results. The extended model indicates an ability to encode the abstract representation of overall sentiment, emotions and sarcasm. A custom dataset was specifically collected for the pre-training objective introduced in this thesis. Future improvements and possible research paths are proposed based on the experiments performed with the extended model.

Abstrakt

Cieľom tejto práce je odprezentovať prehľad aktuálneho výskumu v oblasti analýzy sentimentu bez priameho učenia a identifikovať potenciálne smery výskumu. Okrem toho práca predstavuje novú účelovú funkciu na predtrénovanie, ktorá nevyžaduje priamy supervíziu. Rozšírenie modelu predstavenou účelovou funkciou, pridanie vrstvy neurónovej siete a následné samotné natrénovanie ukazujú sľubné výsledky. Rozšírený model naznačil schopnosť zakódovať abstraktné reprezentácie celkového sentimentu, emócií a sarkazmu. Pre účely použitia predstavenej účelovej funkcie bol nazbieraný vlastný dataset. Na základe experimentov vykonaných s rozšíreným modelom sú odprezentované možné smery výskumu a budúce vylepšenia.

Keywords

sentiment, sentiment analysis, neural network, machine learning, natural language processing, detection, classification

Klíčová slova

sentiment, analýza sentimentu, neurónová sieť, strojové učenie, spracovanie prirodzeného jazyka, dekecia, klasifikácia

Reference

KARABELLY, Jozef. *Non-Supervised Sentiment Analysis*. Brno, 2020. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Martin Fajčík

Rozšířený abstrakt

Vzostup internetu umožnil vznik sociálnych sietí a rôznych verejne dostupných dátových zdrojov. Dnešný svet ponúka množstvo dát a spoločnosti môžu mať veľa dát o zákazníkoch. Avšak analyzovať tieto dáta ručne bez chýb a predsudkov je naozaj pomalé a náročné. Spoločnosti vedia, že potrebujú získať pochopenie dát, ktoré by informovalo ich rozhodnutia. No nemusia vedieť ako takéto pochopenie získať z aktuálne dostupných dát.

Analýza sentimentu poskytuje určitý náhľad do toho, čo je najdôležitejšie zo zákazníckej perspektívy. Okrem toho služby pre zákazníkov nie sú jedinou oblasťou, kde môže byť analýza sentimentu aplikovaná. Ďalšie oblasti sú napríklad obchodovanie na burzách alebo politika.

Keďže analýza sentimentu môže byť prevažne automatizovaná, tak toto znižuje vstup od človeka potrebný počas procesu rozhodovania. Nevýhoda zasahovania človeka do procesu je v tom, že môže zaniesť zaujatosť do procesu. Okrem toho väčšina dnešných najlepších algoritmov vyžaduje veľa označených dát, čo môže byť v niektorých prípadoch problém nie len kvôli zaujatosti, ale aj nedostatku dát.

Pokroky v architektúrach hlbokého učenia v súčasnosti umožnili veľkým architektúram dosiahnuť najlepšie výsledky napriek viacerým NLP úlohám vrátane analýzy sentimentu. Architektúry ako GPT, BERT, XLNet, ELMo, alebo ULMFIT boli predtrénované na obrovských neoznačených datasetoch bez supervízie, vyžadujú menej označených dát na dotrénovanie a objavujú sa na vrcholoch väčšiny porovní v analýze sentimentu.

Analýzou sentimentu bez priameho učenia sa myslia všetky metódy, ktoré nevyžadujú priamu supervíziu. Cieľom analýzy sentimentu bez priameho učenia je dosiahnuť dobré výsledky použitím znalostí zo súvisiacich úloh, iných techník alebo súvisiacich dát.

Práca si dáva za cieľ preskúmať súčasné metódy analýzy sentimentu bez priameho učiteľa, preskúmať potenciálne zlepšenie a navrhnúť smer ďalšieho výskumu. Táto práca popisuje fenomén sentimentu v prirodzenom jazyku a dostupné metódy analýzy sentimentu bez priameho učenia ako napríklad Sentiment Neuron alebo DeepMoji. Vlastná predtrénovacia dátová sada bola zozbieraná z príspevkov na Twitteri a práca predstavuje rôzne porovnávacie dátové sady na klasifikáciu sentimentu, emócií a sarkazmu. Na základe postrehov zo súčasných metód analýzy sentimentu bez priameho učiteľa bola vytvorená nová predtrénovacia účelová funkcia. Prvý postreh je, že úloha na modelovanie jazyka je schopná zachytiť sentiment z textu. Druhý postreh je, že emotikony môžu byť použité ako forma vzdialenej supervízie s výsledkami porovnateľnými s najmodernejšími metódami. Nakoniec sú predstavené rôzne experimenty založené na novej predtrénovacej účelovej funkcii a výsledky sú porovnané voči základnej architektúre.

Základnou architektúrou je DeepMoji, ktorý má zmrazené všetky vrstvy okrem poslednej a tá je dotrénovaná na cieľovej dátovej sade. Celá architektúra bola predtrénovaná na dátach z Twitteru s emotikonmi ako anotáciami. DeepMoji architektúra je diskriminatívna a obojsmerná. Na základe vstupného textu architektúra diskriminuje konktény emotikon. Duplicitné emotikony boli odstránené. Na rozdiel od DeepMoji, Emoji GPT-2 (jedna z architektúr predstavených v experimentoch) je jednosmerná generatívna architektúra, ktorá modeluje pravdepodobnosť celej sekvencie emotikonov.

Na základe výsledkov experimentov je jasné, že celkový sentiment, emócie a sarkazmus môžu byť získané pomocou novej predtrénovacej účelovej funkcie. Jednoduchá LSTM architektúra nebola schopná zachytiť emočný obsah textu ani po predstavený váh pre jednotlivé triedy.

Avšak rozšírená GPT-2 architektúra vyprodukovala porovnateľné výsledky s zakladnou architektúrou ako aj zlepšenie oproti pôvodnej GPT-2 architektúre. Emoji GPT-2 architek-

túra prekonala základnú DeepMoji architektúru iba v jednom porovnaní úlohy klasifikácie emocií. Napriek tomu výsledky porovnaní z úlohy klasifikácie sentimentu a sarkazmu boli porovnateľné. A čo viac Emoij GPT-2 je založená na predtrénovanom GPT-2, takže vyžaduje menej trénovacieho času a dát na dosiahnutie podobných výsledkov.

Použitie architektúry založenej na Transformer architektúre otvorilo možnosti k analýze pozornosti. Zaujímavá skupina pozorností bola objavená počas analýzy pozornosti v Emoji GPT-2. Bolo by zaujímavé vidieť možnosť klasifikácie sentimentu iba na základe pozornosti.

Záverom tejto práce na základe experimentov je potvrdené, že modelovanie jazyka je schopné extrahovať sentiment z textu. Taktiž predstavená účelová funkcia je schopná zachytiť lepšie reprezentácie emócií než skryté reprezentácie z Transformer architektúry skoro o 5 percent na niektorých úlohách. Taktiež emotikony ako foma vzdialenej supervízie sú výborné v zachytávaní sentiment, rôznych emócií ako aj sarkazmu. Potenciálne vylepšenia tejto práce do budúcna sú prieskum LSTM architektúry, výskum klasifikácie založenej na pozornosti a porovnanie výsledkov s architektúrou dotrénovanou na špecifických dátových sadách.

Non-Supervised Sentiment Analysis

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Ing. Martin Fajčík. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....
Jozef Karabelly
May 28, 2020

Acknowledgements

I would like to thank my supervisor Ing. Marting Fajčík for his exemplary guidance and constructive criticism. He was always willing to answer any questions about my research or writing. He steered me in the right direction anytime I needed it. This thesis would not have been possible without him. Thank you.

Contents

Chapter 1

Introduction

The rise of the Internet caused the creation of social media and many different publicly available data sources. Today's world offers a lot of data, and companies might have a lot of customer data. However, it is really slow and hard to analyze the data manually by a person without any errors or bias. Companies know insight is needed to inform their decisions. However, they might not know how to get insight from the available data.

Sentiment analysis provides some insight into what is most important from the customer perspective. Furthermore, customer service is not the only area where sentiment analysis can be applied. Others are, for example, stock market trading or politics. Leading up to the Great Britain/European Union Membership Referendum (Brexit), a sentiment analysis tool was able to predict around six hours before the announcement that polls favouring the 'remain' camp were incorrect. Figure 1.1 shows sentiment analysis tool monitoring and measuring sentiment from social media posts during the Brexit. The polling stations closed at 22:00 and the tool predicted the result around 16:00 (see Figure 1.1).

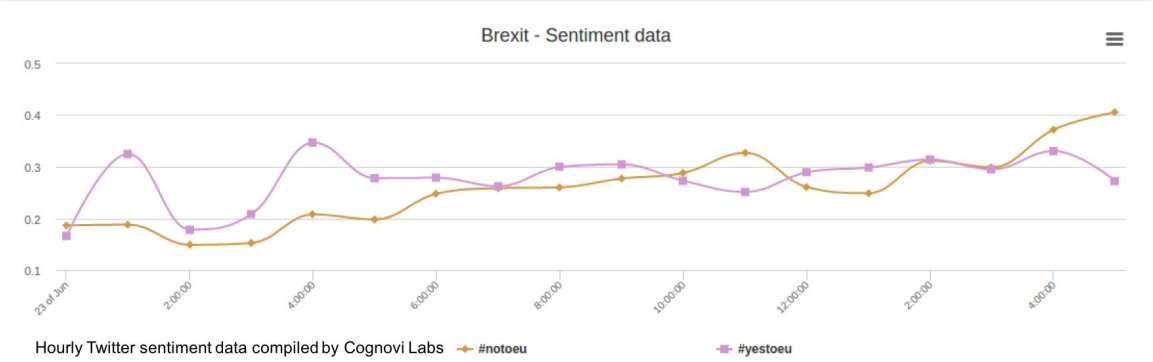

Figure 1.1: The example of sentiment analysis tool.[7] The figure shows levels of sentiment for both camps in the Brexit polls. The polling stations closed at 22:00 and the tool started to predict the correct outcome already around 16:00 as indicated in the last time point in figure.

Since the sentiment analysis can be mostly automated, it alleviates most of the human intervention during the decision making process. There is a downside to human intervention due to bias and subjective judgement. Nevertheless, most of the current best-performing

algorithms require a lot of labelled data which can be in some cases a problem not only due to bias but also due to data scarcity.

The advancements in deep learning architectures in recent years enabled large architectures to achieve the state of the art results across many NLP tasks, including sentiment analysis. Architectures like GPT [30], BERT [12], XLNet [45], ELMo [28] or ULMFIT [17], pre-trained on large unlabelled datasets without supervision, require less labelled data for fine-tuning and appear on the top of most sentiment analysis benchmarks.

Non-supervised sentiment analysis refers to methods that do not need direct supervision and the goal of the non-supervised sentiment analysis is to achieve good results using knowledge from related tasks, other techniques or related data.

The goal of the thesis is to explore the current state of non-supervised sentiment analysis methods, explore potential improvements and to give suggestions for the future research. The thesis is structured in the following manner. Chapter 2 describes the different types of sentiment analysis and used methods. The key concepts needed to understand methods that are described in later chapters are in Chapter 3. Chapter 4 contains some of the current non-supervised sentiment analysis methods. There are various architectures to create an overview of different approaches from the field. The dataset specifically collected for the thesis and different benchmark datasets are described in Chapter 5. Chapter 6 details models created and used in the thesis and the experiments performed with them. The experiments and the model results are evaluated in Chapter 8.

Chapter 2

Sentiment Analysis

The sentiment [35] is a thought, opinion, or idea based on a feeling about a situation, or a way of thinking about something. However, it is also defined as gentle feelings such as sympathy, love, etc.
Due to its ambiguity, the task of analyzing the sentiment is further divided into multiple areas such as emotion detection or polarity classification.

Sentiment analysis refers to a problem of systematically extracting subjective information from a text and classifying that information into multiple categories.

Moreover, sentiment analysis can be performed at various levels of textual granularity:

- Document-level where the task is to detect the overall sentiment of a whole document or a paragraph.
- Sentence-level where the task is to detect the sentiment of the given sentence.
- Subsentence-level where the task is to detect the sentiment of sub-structures within the sentence, eg. words.

2.1 Different types of sentiment analysis

There are various types of sentiment analysis ranging from basic polarity classification (positive, neutral) to detection of feelings and emotions (anger, fear, sadness) or subjectivity/objectivity classification.

Polarity	Sentence Sample
positive	Béart and Berling are both superb, while Huppert ... is magnificent.
positive	Not only does Spider-Man deliver, but I suspect it might deliver again.
negative	Without Shakespeare's eloquent language, the update is dreary and sluggish.
negative	Final verdict: You've seen it all before.

Table 2.1: Example of polarity classification on IMDB reviews.

2.1.1 Fine-grained sentiment analysis

Sentiment analysis can be as simple as binary positive/negative opinion polarity classification. However, there can be a need for more opinion categories to consider: Very positive, Positive, Neutral, Negative, Very negative.

This type is generally called fine-grained sentiment analysis and might be, for example, used to classify 5-star rating reviews.

For more information about this task, refer to [11].

Polarity	Sentence
very positive	Still, this flick is fun, and host to some truly excellent sequences.
positive	Yet the act is still charming here.
neutral	Some movies blend together as they become distant memories.
negative	This isn't a new idea.
very negative	It's not a motion picture; it's an utterly static picture.

Table 2.2: Example of fine-grained sentiment analysis from Stanford Sentiment Treebank [38].

2.1.2 Emotion detection

Emotion detection is the process of classifying emotions expressed in the text. Most common theories of emotions used are Ekman's six basic emotions [13] along with Plutchik's wheel of emotions [29], which describes eight basic emotions. However, some words that would normally express anger might express joy too. For example, the word kill can be used as an expression of anger, e.g. you are killing me, but also a joy, e.g. you killed it.

For additional information about this task see [36].

Emotion	Topic sentences
Neutral	Warlords of Draenor Launch Update: 1:40 p.m.
Joy	Thanks, Blizzard!
Trust	Female Gamers Group want a Safe Space.
Fear	What's going on with the WoW Armory?
Surprise	Why no-flying in Draenor is a good thing!
Sadness	Server Maintenance soo...rate that xmog!
Anger	Curse client not connecting to the internet?
Anticipation	The new models aren't final, right?

Table 2.3: Example of emotion detection from MMORPG game.

2.1.3 Subjectivity classification

The task of subjectivity classification is to determine whether a text expresses an opinion or not, in other words, classify if a given text is objective or subjective. Afterwards, in the case of subjective information, the goal is to classify polarity of the given opinion. However, the problem is that many words have both subjective and objective meaning. For example, the word positive is objective when its meaning is electropositive (e.g. protons are positive),

but it is subjective when used to express advantage (e.g. a positive factor). Furthermore, the objective text might contain subjective sentences (e.g. quotes in news articles).

More on this task in [40].

Polarity	Subjectivity	Sample Sentence
Negative	Subjective	Literally ur facebook message app is useless
Positive	Subjective	The battery life of this camera is very good
-	Objective	Camera is a good device for capturing photographs

Table 2.4: Example of subjectivity classification on tweets.

2.1.4 Aspect-based sentiment analysis

Traditionally, sentiment analysis classifies the overall sentiment of a document. Aspect-based sentiment analysis task is defined as classifying polarity of the opinions on different aspects expressed in a text. The aspects are attributes or components of a subject of interest. The advantage is that aspect-based analysis might capture more information about a subject of interest. Different aspects can have different sentiment, for example, an airline can provide great comfort, but bad food.

Table 2.5: Example of aspect-based sentiment analysis on airlines review

Aspect	Sentiment	Sample Sentence
food	negative	What I will complain about is the food on offer....
punctuality	positive	My flight arrived on time which was great....
comfort	negative	Also, the seats are a uncomfortable for people lik...
value	neutral	The price I paid was really cheap in comparison to...
staff	positive	It's a cheap A-B service that saves you a tonne of...

2.2 Sentiment analysis methods

The approaches to sentiment analysis that can be grouped into these three categories: knowledge-based, statistical or hybrid methods.

2.2.1 Knowledge-based methods

These methods classify text into affect categories (e.g. anger, joy, fear) based on the presence of unambiguous affect words (e.g. happy, sad, afraid, and bored), phrases and phrase patterns [5]. However, knowledge-based techniques are problematic in the following two areas:

- Cannot reliably recognize affect-negated words. For example, sentence 'that meal was excellent' might be correctly classified as positive, however, it is also likely to assign the same class to the sentence 'that meal wasn't excellent'.
- Reliance on surface features, in other words, a sentence sometimes expresses affect by meaning, rather than affect adjectives.

2.2.2 Statistical methods

These methods use elements from statistics or machine learning to build a model. By training a model on a large training corpus of documents with affect annotations, the model might not learn just affect of the keywords, but also, for instance, take into consideration contextual features like arbitrary keywords, punctuation, and word co-occurrence frequencies.

Generally used are neural networks based loosely on functions of the human brain. The neural network approximates a function which maps inputs x to outputs y. For natural language processing tasks, generally, the input is a vector representing a text or a word, and the output are probabilities from a probability distribution of classes C $P(C = c|x)$. Specific methods used in sentiment analysis are described later in Chapter 4.

The input text is generally transformed by different approaches into numerical representation, usually vector. Sometimes, this vector represents word or expression frequencies in a predefined dictionary, and the common method has been the bag of words with their frequencies [15]. This approach looks at the histogram of the words within the text, i.e. considers each word count as a feature.

However, newly used feature extraction techniques utilize word embeddings, also called word vectors. The feature vector represents different aspects of the word: each word is associated with a point in a vector space. The number of features is significantly smaller than the size of the vocabulary and the words with similar meaning have similar representations which might improve the performance of the classifier [2]. The frequently used method used is word2vec [23].

2.2.3 Hybrid methods

Hybrid methods utilise both machine learning and knowledge-based methods like ontologies and semantic networks. These allow extracting conceptual and affective information related to natural language opinions. These methods no longer rely on blindly using keywords, however, on large semantic knowledge bases. The advantage is that they can recognise the sentiment that is expressed subtly. On the other hand, the disadvantage is their reliance on the depth and breadth of the used knowledge bases [5].

Hybrid methods showed promise at fine-grained feature-based sentiment analysis [6].

2.3 Evaluation

Various performance metrics are used to evaluate a classifier and to understand a sentiment analysis system. Traditional evaluation metrics of a classifier performance are precision, recall, F1 and accuracy explained later in this section.

2.3.1 Confusion matrix

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Table 2.6: Confusion matrix

A confusion matrix outlines the performance of a classifier over a set of data samples. It is a two-dimensional matrix, indexed in one dimension by the true class of an object and in the second by the class assigned by the classifier. A confusion matrix has two classes, one labelled as the positive class and the other the negative class. In this regard, the four cells of the matrix are labelled: true positives (TP), false positives (FP), true negatives (TN), false negatives (FN), as illustrated in Table 2.6.

2.3.2 Accuracy

The accuracy of a model is usually evaluated by applying it to test data for which the labels are known. The accuracy of a classifier on test data might be calculated as:

$$Accuracy = \frac{\text{the number of correctly classified objects}}{\text{total number of objects}} \tag{2.1}$$

Using accuracy as the main performance metric with a serious class imbalance can be a problem. For instance, consider a problem where only 1% of the examples belong to the positive class, high accuracy of 99% is achievable by predicting the negative class for all examples. However, all positive class examples, the rare and more interesting cases, are misclassified.

To give a better insight into the performance of a model, precision, recall and F1 score should be considered too.

2.3.3 Precision

Precision is defined as the ratio of true positive (TP) and the total number of positive predictions by a model. This is defined regarding confusion matrix with two classes as mentioned above. Precision can then be defined in terms of true positives and false positives (FP) as follows.

$$precision = \frac{TP}{TP + FP} \tag{2.2}$$

Precision is useful when the penalty for false positives is too high, for example, in models for cancer detection.

2.3.4 Recall

The recall is a measure of information extraction performance. It is also defined regarding confusion matrix, similarly as precision, and it is related to precision. Recall can be defined in terms of true positives and false negatives (FN) as follows.

$$recall = \frac{TP}{TP + FN} \tag{2.3}$$

The recall is useful when the penalty for false negative is too high, for example, in models for nuclear missile retaliation.

The tradeoff is that increasing recall decreases precision, and vice versa while keeping the model same. For example, this might be done by changing the threshold (see Figure 2.1). There is a whole scientific field behind finding an optimal threshold called „Decision Theory“.

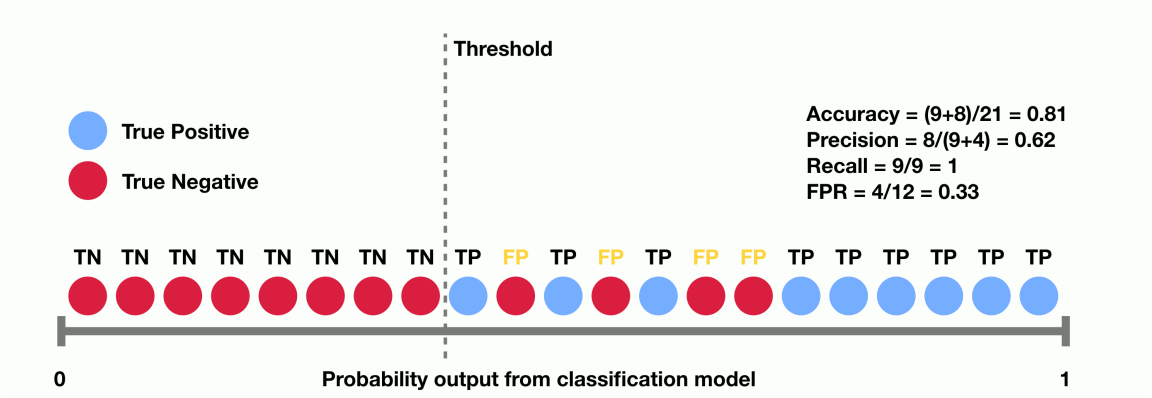

Figure 2.1: Threshold example [9].

2.3.5 F1 score

F1-score is a measure of the accuracy of predictions in binary classification problems. It is defined as the harmonic mean of precision and recall (see Equation 2.2 and Equation 2.3). F1-score is defined as follows.

$$F1 = 2 * \frac{(precision * recall)}{(precision + recall)} \tag{2.4}$$

Therefore, a system with high F1-score has both good precision and good recall. The F1-score is also used to evaluate multiclass classification problems. In this case, the final score is calculated by micro-averaging or macro-averaging (see Equation 2.7 and Equation 2.10 respectively). Micro-average sums the contributions (TP, FP, TN, FN) for all classes C to calculate the average score. Macro-average calculates the F1-score independently for each class and then computes the average, which means it treats all classes equally and does not take label imbalance into account.

$$P_{micro} = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + FP_i} \tag{2.5}$$

$$R_{micro} = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + FN_i} \tag{2.6}$$

$$F1_{micro} = 2 * \frac{P_{micro} * R_{micro}}{P_{micro} + R_{micro}} \tag{2.7}$$

The micro-averaged precision P_{micro} and R_{micro} are calculated, and then used to calculate micro-averaged $F1_{micro}$.

$$P_{macro} = \frac{\sum_{i=1}^{C} P_i}{C} \tag{2.8}$$

$$R_{macro} = \frac{\sum_{i=1}^{C} R_i}{C} \tag{2.9}$$

$$F1_{macro} = 2 * \frac{P_{macro} * R_{macro}}{P_{macro} + R_{macro}} \tag{2.10}$$

2.4 Inter-annotator reliability

Inter-annotator reliability is the degree to which are two or more annotators in agreement. It solves the problem of consistency in rating systems implementation. High inter-annotator reliability values reflect a high degree of agreement between two annotators. Low inter-annotator reliability values reflect a low degree of agreement between two annotators.[21] One of the most frequently used for inter-reliability is Krippendorff's alpha (α).

α's general form is:

$$\alpha = 1 - \frac{D_o}{D_e} \tag{2.11}$$

where D_o is the observed disagreement among values assigned to units of analysis and D_e is the disagreement one would expect when the codingof units is attributable to chance.

For example, the best inter-rater agreement for Twitter sentiment analysis hits 0.655 with Krippendorff's alpha [33]. This means there is a good degree of agreement, however, it is still far from ideal 0.8 which is what social scientists commonly rely on.[20]

Chapter 3

Key Concepts

3.1 KL divergence

Kullback-Leibler Divergence is an asymmetric measure of the difference between two distributions [4]. Original probability distribution P and approximate distribution Q.

$$D_{KL}(P||Q) = \sum_{i=1}^{N} P(x_i)(\log P(x_i) - \log Q(x_i))$$

$$D_{KL}(P||Q) = \mathbb{E}[\log P(x) - \log Q(x)] = \mathbb{E}_{P(x)}[P(x)] - \mathbb{E}_{P(x)}[Q(x)]$$

KL divergence calculates how much information is lost when one distribution is approximated with another.

3.2 Likelihood function

The likelihood function measures how well the data describes the unknown parameters of the model. The unknown parameters of a distribution are denoted as θ and the data as X. The probability density function can be denoted as $f(x|\theta)$ since the probability distribution depends on the parameters. Furthermore, when $X = x$ is the observed sample data point, then the function of θ defined as:

$$L(\theta|x) = f(x|\theta)$$

is the likelihood function. The difference between the likelihood function and the PDF is which variable is considered fixed and which is changing. In the case of the PDF, the x is the variable, and θ is fixed. However, the x is observed sample point in the likelihood function, and the θ varies over all possible parameter values. Furthermore, although $f(x|\theta)$, as a function of x, is a PDF, there is no guarantee that $L(\theta|x)$, as a function of θ, is a PDF.

3.3 LSTM

Long Short Term Memory [16] networks are a special kind of RNN, capable of remembering longer contexts than basic RNN. The key to the LSTMs is the cell state represented by the top horizontal line in Figure 3.1. The addition and removal of the information in the cell state are controlled by gates. Gates let the information through and consist of a sigmoid

function and a pointwise multiplication operation.

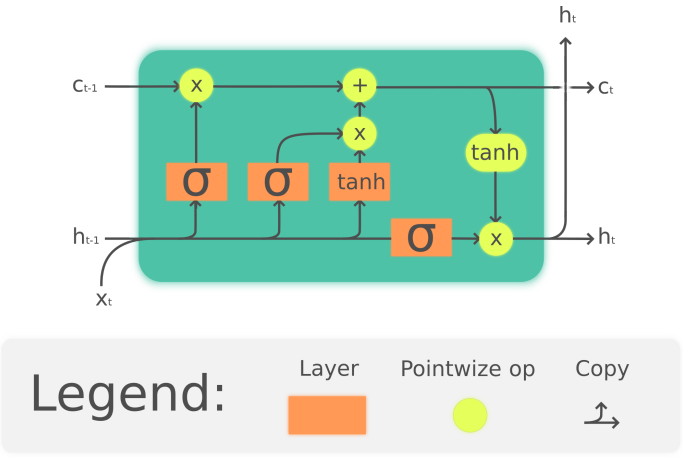

Figure 3.1: LSTM cell [8].

There are three types of these gates, forget gate f_t, input gate i_t and output gate o_t.
The forget gate 'decides' what information is going to be thrown away from the cell state by looking at h_{t-1}, information from the previous step, as shown in Equation 3.1.
The input gate determines what new information is going to be stored in the cell state, as shown in Equation 3.2. In vector $\tilde{C}_t$ is the new information that could be added to the state C_t.
Lastly, the output gate filters the information, as shown in Equation 3.3, that is going to be the final output h_t.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{3.1}$$
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{3.2}$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{3.3}$$
$$\tilde{C}_t = \tanh(W_C x_t + U_C h_{t-1} + b_C) \tag{3.4}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{3.5}$$
$$h_t = o_t \odot \tanh(C_t) \tag{3.6}$$

The $\odot$ denotes the Hadamard product (element-wise product), which takes two matrices A and B with the same dimensions and produces a matrix C with the same dimensions where the elements are calculated as shown in Equation 3.7.

$$(C)_{ij} = (A \odot B)_{ij} = (A)_{ij}(B)_{ij} \tag{3.7}$$

On the other hand, the matrix product (dot product) takes two matrices A and B such that A is an $m \times n$ matrix and B is an $n \times p$ matrix and produces $m \times p$ matrix C where the elements are calculated as shown in Equation 3.8.

$$(C)_{ij} = \sum_{k=1}^{n} (A)_{ik}(B)_{kj} \tag{3.8}$$

3.4 mLSTM

mLSTM [19] combines the long short-term memory (LSTM), see Section 3.3, and multiplicative recurrent neural network architecture. These architectures are combined by adding connections from mRNN's intermediate state m_t to each gating unit in the LSTM as follows:

$$m_t = (W_{mx}x_t) \cdot (W_{mh}h_{t-1}) \tag{3.9}$$

$$\hat{h}_t = W_{hx}x_t + W_{hm}m_t \tag{3.10}$$

$$i_t = \sigma(W_{ix}x_t + W_{im}m_t) \tag{3.11}$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_t) \tag{3.12}$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_t) \tag{3.13}$$

The dimensionalities of m_t and h_t are equal and m_t is shared across all LSTM unit types, meaning that m_t interacts additively across all the units. This architecture combines the flexible input-dependent transitions of mRNN and the long memory and information control of LSTM. The extra sigmoid input and forget gate featured in LSTM enable better input-dependent transition functions than in regular mRNN. Also, it is faster than LSTM.

3.5 Language modelling

Language Modelling (LM) is one of the numerous essential parts of modern Natural Language Processing (NLP). Language modelling is the task of assigning a probability to sequences in a language. Apart from assigning a probability to each sequence of words, the language model assigns a probability for a given word or a sequence of words to follow a sequence of words.[15] Formally, the task of language modelling is to estimate the probability of a sequence of words $P(w_1, .., w_n)$, which in practice is usually rewritten using the chain rule of probability as:

$$P(w_1, .., w_n) = P(w_1)P(w_2|w_1)...P(w_n|w_1, ..., w_{n-1}) \tag{3.14}$$

3.6 Byte Pair Encoding

Most common word representations cannot handle unseen or rare words well. Character-level embeddings are one of the solutions to out-of-vocabulary words. However, character-level language units might be too fine-grained to capture some important information. Subword level is between character and word, which means it is not as fine-grained as character level. Furthermore, it handles unseen and rare words.

Byte Pair Encoding (BPE) [34] is a simple data compression technique that iteratively replaces the most frequent pair of bytes in a sequence with a single, unused byte. This technique is used for word segmentation. However, instead of merging frequent pairs of bytes, it merges characters or character sequences.

The algorithm, shown in Figure 3.2, is as follows:

1. Initialize symbol vocabulary, where each word is represented as a sequence of characters plus a special end-of-word symbol, <\w> in Figure 3.2.

2. Generate a new subword based on the high occurrence frequency.

3. Repeat 2. until the final symbol vocabulary size is equal to the size of the initial vocabulary plus the number of merge operations, which is the only hyperparameter of the algorithm.

Algorithm 1 Learn BPE operations

```
import re, collections

def get_stats(vocab):
  pairs = collections.defaultdict(int)
  for word, freq in vocab.items():
    symbols = word.split()
    for i in range(len(symbols)-1):
      pairs[symbols[i],symbols[i+1]] += freq
  return pairs

def merge_vocab(pair, v_in):
  v_out = {}
  bigram = re.escape(' '.join(pair))
  p = re.compile(r'(?<!\S)' + bigram + r'(?!\S)')
  for word in v_in:
    w_out = p.sub(''.join(pair), word)
    v_out[w_out] = v_in[word]
  return v_out

vocab = {'l o w </w>' : 5, 'l o w e r </w>' : 2,
         'n e w e s t </w>':6, 'w i d e s t </w>':3}
num_merges = 10
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
  print(best)
```

Figure 3.2: The minimal Python implementation of BPE. [34]

3.7 Transformer architecture

When a sequence is processed by LSTM, as described in 3.3, each hidden state depends on the previous hidden states. This makes LSTM, and recurrent models as a whole, inefficient and slow even on GPUs, as the temporal dependencies are unlikely to be parallelized.

Another problem of LSTMs is learning long-range dependencies by a network. Theoretically, LSTMs can have long-term memory, however, remembering long-range dependencies is still a challenge due to gradient vanishing/explosion [26]. Furthermore, some words have different meanings based on the context.

The traditional attention mechanism improved the solution to the problem of long temporal dependencies between the input and output tokens. The idea behind the Transformer is to extend this mechanism to the input and output sentence processing, which means the sequence encoder and decoder can see the entire input sequence and generated output sequence all at once.

The Transformer [42] follows encoder-decoder architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder, shown in the left and right halves of Figure 3.3.

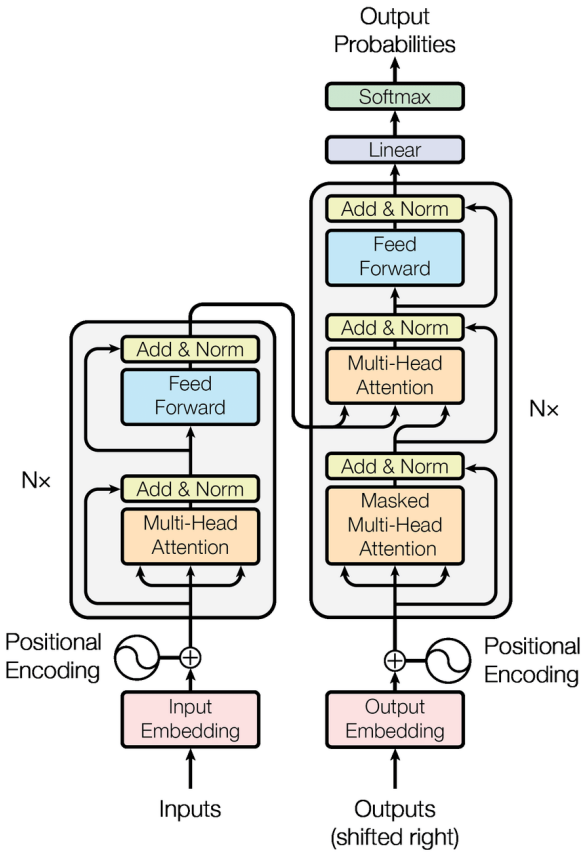

Figure 3.3: The Transformer - model architecture. [42]

Each encoder layer l has two sub-layers. The first is a multi-head self-attention mechanism $\tilde{h_i}$, and the second is a simple position-wise fully connected feed-forward network $\tilde{x_i}$. Between each sub-layer is a residual connection followed by layer normalization [1] $LayerNorm$.

$$h_i^{\tilde{l+1}} = \sum_{m=1}^{M} W_m^{l+1}[\sum_{j=1}^{N} A_{i,j}^m \cdot V_m^{l+1} x_j^l] \tag{3.15}$$

$$h_i^{l+1} = LayerNorm(x_i^l + h_i^{\tilde{l+1}}) \tag{3.16}$$

$$x_i^{\tilde{l+1}} = W_2^{l+1} \cdot RELU(W_1^{l+1} h_i^{l+1} + b_1^{l+1}) + b_2^{l+1} \tag{3.17}$$

$$x_i^{l+1} = LayerNorm(h_i^{l+1} + x_i^{\tilde{l+1}}) \tag{3.18}$$

The m is the attention head index and $A_{i,j}^m$ represents attention weights between elements i and j. The Equation 3.17 uses $RELU$ activation function [24].

The decoder is similar to the encoder. In addition to the two sub-layers in each layer, there is a third sub-layer performing multi-head attention over the output of the encoder stack. Furthermore, the self-attention sub-layer is modified to mask inputs to the decoder from the future timesteps.

3.8 GPT-2

GPT-2 [32] is a large transformer-based language model (architecture described in Section 3.7) trained on a dataset of 8 million web pages. The GPT-2 is built using transformer decoder blocks as opposed to BERT, which uses transformer encoder blocks, architecture shown in Figure 3.4.

However, the key difference from BERT is that GPT-2 was trained with a causal (unidirectional) language modelling objective. Therefore, it is powerful at predicting the next token in a sequence and utilizing this feature allows GPT-2 to generate syntactically coherent text. The input sequences are encoded by byte-level Byte Pair Encoder, the algorithm described in Section 3.6, into tokens and then transformed to vector embeddings.

The last decoder layer outputs are used for classification. The pre-training process is expensive, however, done only once, and then fine-tuned for specific tasks.

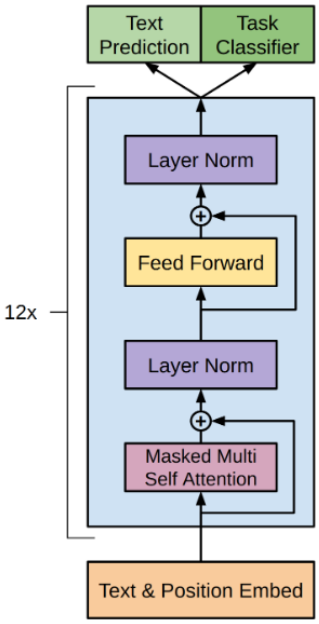

Figure 3.4: Transformer architecture and training objectives used in GPT. [30]

3.9 Measures of association

The Cramér's V [10] is a measure of association between two nominal variables where the output of the method is a value between 0 and 1 inclusive, and it is based on Pearson's chi-squared test. It is a symmetrical method meaning the order of the input variables does not matter. The original Cramér's V suffers from bias, so the formula with the bias correction [3] was used, shown in Equation 3.19.

$$\tilde{k} = k - \frac{(k-1)^2}{n-1} \tag{3.19}$$

$$\tilde{r} = r - \frac{(r-1)^2}{n-1} \tag{3.20}$$

$$\tilde{\varphi} = max(0, \varphi^2 - \frac{(k-1)(r-1)}{n-1}) \tag{3.21}$$

$$\tilde{V} = \sqrt{\frac{\tilde{\varphi}^2}{min(\tilde{k}-1, \tilde{r}-1)}} \tag{3.22}$$

It is calculated by taking the square root of the corrected phi coefficient $\tilde{\varphi}$ divided by the corrected minimum dimension minus 1. The n is the total number of observations, and k and r are the numbers of unique values from each input variable.

The Theil's U also referred to as uncertainty coefficient is also a measure of association, and it is based on the concept of information entropy. This measure lies between 0 and 1 similarly to the Cramér's V. However, it is not symmetric concerning the two input variables, and this prevents information loss. The formula for Theil's U is shown in Equation 3.23.

$$U(X|Y) = \frac{H(X) - H(X|Y)}{H(X)} \tag{3.23}$$

It is calculated as the entropy of variable X minus conditional entropy of X given Y divided by the entropy of X. The conditional entropy is calcualted as:

$$H(X|Y) = -\sum_{x,y} P_{X,Y}(x,y) \log P_{X|Y}(x|y) \tag{3.24}$$

where the $P_{X,Y}(x,y)$ is joint distribution and $P_{X|Y}(x|y)$ conditional distribution.

Chapter 4

Non-supervised sentiment analysis methods

This chapter describes a different state of the art methods used for non-supervised sentiment analysis.

4.1 Variational approach with target-opinion word pairs

Document-level multi-aspect sentiment analysis aims to predict the sentiment polarity of each aspect given a text which comprises several sentences containing one or more aspects. This method [46] uses target-opinion word pairs as a self-supervision. For example, in a document "The bedroom is very spacious," extracted target-opinion pair "bedroom-spacious" might indicate that the sentiment polarity of the aspect room is positive.

These word pairs can be extracted using rule-based methods and used in the model which consists of a sentiment polarity classifier and an opinion word classifier. Both classifiers are trained for each aspect separately. The input of the sentiment classifier of each aspect, i.e., a representation of a document, is the same. The target-opinion word pairs passed to opinion word classifiers are different for various aspects. The connection between these two classifiers is shown in Figure 4.1.

The input x is a representation of a document and is passed into the sentiment polarity classifier which produces a distribution of the sentiment polarity R_a for an aspect a, denoted as $q(R_a|x)$. When R_a has only two values, positive and negative, then the sentiment classifier outputs are $q(positive|x)$ and $q(negative|x)$. The opinion word classifier takes a target word and a possible value of the sentiment polarity r_a as input and estimates $p(w_o|r_a, w_t)$ where w_o is opinion word representation and w_t is target word representation.

The objective function is to maximize a log-likelihood of an opinion word w_o, described in 3.2, given a target word w_t. Also as mentioned earlier, the objective function can be rearranged into two sub-tasks. The first corresponds to sentiment polarity classifier and the second to the opinion classifier. A variational lower bound of the log-likelihood, which includes both classifiers, can be derived after addition of a latent variable - the sentiment polarity, as follows:

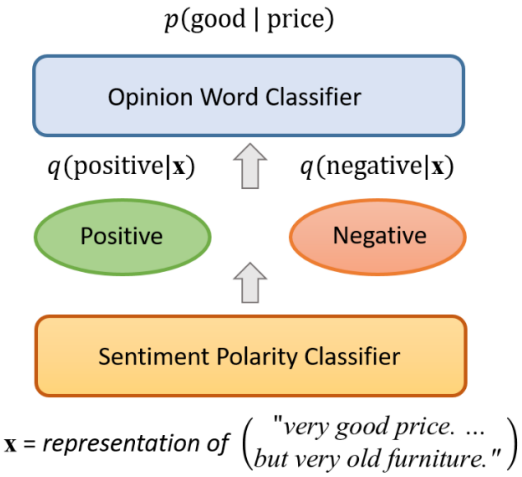

Figure 4.1: A sentiment polarity classifier and an opinion word classifier associated with the aspect *price*.[46]

$$
\begin{aligned}
\mathcal{L} &= \log p(w_o|w_t) \\
&= \log \sum_{r_a} p(w_o|r_a, w_t) \\
&= \log \sum_{r_a} q(r_a|x)[\frac{p(w_o, r_a|w_t)}{q(r_a|x)}] \\
&\geq \sum_{r_a} q(r_a|x)[\log \frac{p(w_o, r_a|w_t)}{q(r_a|x)}] \\
&= \mathbb{E}_{q(R_a|x)}[\log p(w_o|r_a, w_t)p(r_a|w_t)] + H(q(R_a|x)) \\
&= \mathbb{E}_{q(R_a|x)}[\log p(w_o|r_a, w_t)p(r_a)] + H(q(R_a|x)),
\end{aligned}
\tag{4.1}
$$

where $H(\Delta)$ refers to the Shannon entropy and after application of Jensen's inequality, the log-likelihood is lower-bounded. The equality is true if and only if the KL-divergence, described in 3.1, of two distributions, $q(R_a|x)$ and $p(R_a|w_t, w_o)$, equals to zero. Maximisation of the lower bound is equivalent to the minimisation of the KL-divergence. Therefore, a sentiment polarity classifier can be trained to produce a similar distribution to the true posterior $p(R_a|w_t, w_o)$. $q(R_a|x)$ is more flexible, compared with $p(R_a|w_t, w_o)$, because it takes any kind of document representation as input. There are two assumptions made. Firstly, a target word w_t is independent of a sentiment polarity and vice versa since the polarity assignment is not affected by the target word. Secondly, the sentiment polarity R_a follows a uniform distribution, which means $p(r_a)$ is a constant and it can be removed from the objective function as follows.

$$
\mathbb{E}_{q(R_a|x)}[\log p(w_o|r_a, w_t)] + H(q(R_a|x)) \tag{4.2}
$$

This approach can achieve similar results to the supervised method with hundreds of labels per aspect, which can reduce a lot of labour work in practice.

4.2 Emoticons as distant supervision

A range of NLP tasks is limited by an insufficient amount of manually annotated data. In consequence, co-occurring emotional expressions were used for distant supervision in social media sentiment analysis and associated tasks to cause the models to learn good text representations before modelling these tasks. Distant supervision on noisy labels allows a model to achieve better performance on the target task. Emoticons do not always represent a direct label of emotional content. Nonetheless, emojis can be utilised to classify emotional content of text accurately in several cases [14].

The model uses an embedding layer of 256 dimensions, which projects each word into vector space and a hyperbolic tangent activation function (see Equation 4.3). It is continuous on its domain $D(tanh) = \mathbb{R}$, and limits at endpoints of the domain are $\lim_{x\to-\infty} tanh(x) = -1$ and $\lim_{x\to\infty} tanh(x) = 1$.

$$tanh(x) = \frac{\exp^{x} - \exp^{-x}}{\exp^{x} + \exp^{-x}} \tag{4.3}$$

There are two bidirectional LSTM layers, the architecture described in Section 3.3, with 1024 hidden units in each (512 in each direction) to capture the context. Lastly, all layers are passed into the attention layer using skip-connections, as illustrated in Figure 4.2.

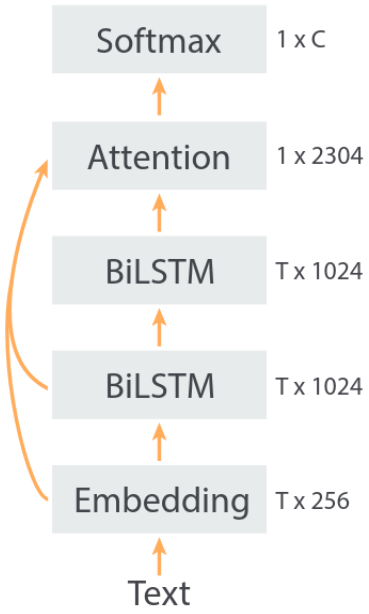

Figure 4.2: Illustration of the model where T is text length and C the number of classes.[14]

The attention mechanism enables the model to decide the weight of each input token for the prediction task when creating the representation. The model uses a simple approach with a single parameter per input channel:

$$
\begin{aligned}
e_t &= h_t^\top w_a \\
a_t &= \frac{\exp(e_t)}{\sum_{i=1}^{T} \exp(e_i)} \\
v &= \sum_{i=1}^{T} a_i h_i
\end{aligned}
\tag{4.4}
$$

where h_t is the representation of the word at time step t and w_a is the weight vector for the attention layer. The attention importance scores a_t for each time step are the result of multiplication of the representations with the weight vector and then normalized to create a probability distribution over the tokens. Finally, the representation vector v of the text is calculated by a weighted summation over all the time steps using the attention importance scores as weights. This representation vector is a high-level encoding of the entire text, which is used as the input for the final layer for classification.

Many of the emoji have similar emotional content, however, they have slight differences in usage that the model captures. The agglomerative hierarchical clustering on the correlation matrix of the predictions shows similarities captured by the model. As illustrated in Figure 4.3, the model groups emojis into overall categories associated with e.g. negativity, positivity or love.

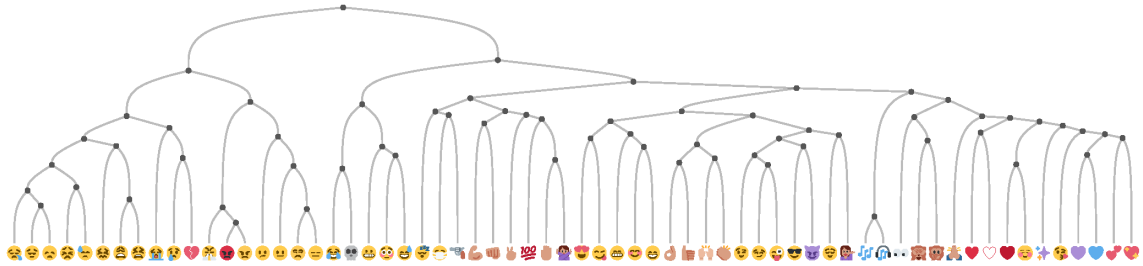

Figure 4.3: Hierarchical clustering of the DeepMoji model's predictions across categories on the test set. The dendrogram shows how the model learns to group emojis into overall categories and subcategories based on emotional content. The y-axis is the distance on the correlation matrix of the model's predictions measured using average linkage, the distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster.[14]

This approach yields state-of-the-art performance on benchmark datasets within sentiment, emotion and sarcasm detection using a single pre-trained model. The model is used as a baseline for the proposed method.

4.3 Sentiment neuron in the recurrent language model

This method [31] utilizes the properties of recurrent language models. It considers byte-level language modelling due to its simplicity and generality. Due to a small vocabulary, the model is compact. Furthermore, since it operates on a byte level rather than on language-specific words or characters, it can analyze a text in different languages. The language model is trained on a very large corpus and a single layer multiplicative LSTM with 4096

units. It has been chosen because multiplicative LSTMs converge faster than normal LSTMs for the chosen hyperparameters. The model processes input text as a sequence of UTF-8 encoded bytes. It updates its hidden state and predicts a probability distribution over the next possible byte for each byte. The hidden state of the model represents the context of the sequence which contains all information the model learnt is relevant to predict the future bytes of the sequence.

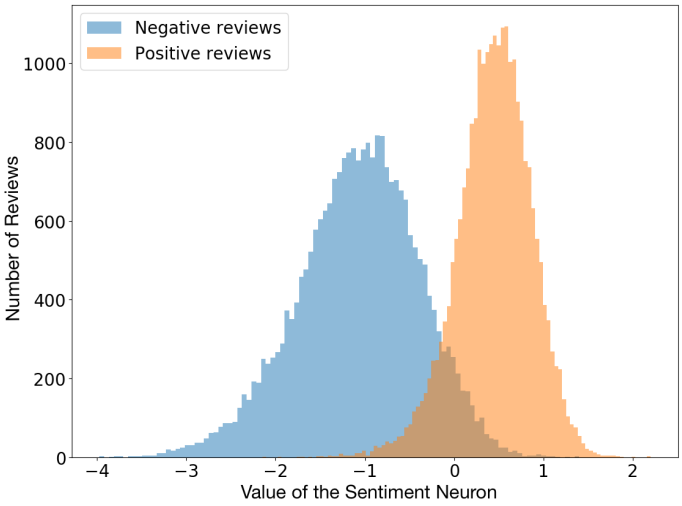

Figure 4.4: Histogram of cell activation values for the sentiment unit on IMDB reviews.[31]

After inspecting the contributions of features on different datasets, there is a single unit within the mLSTM that directly corresponds to sentiment. The histogram of activations of this unit from the final hidden state, as illustrated in Figure 4.4, shows a clear separation between positive and negative reviews on IMDB dataset. The visualization of the activations of this unit on 6 randomly selected reviews from a set of 100 high contrast reviews shows an on-line estimate of the local sentiment, shown in Figure 4.5.

It is an open question of why this model recovers the concept of sentiment in such a precise, disentangled, interpretable, and manipulable way. Maybe sentiment as a conditioning feature has a strong predictive capability for language modelling. However, the model is sensitive to the data it was trained on. It is unrealistic to expect a model trained on a corpus of books to learn an encoding which preserves the exact sentiment of a review.

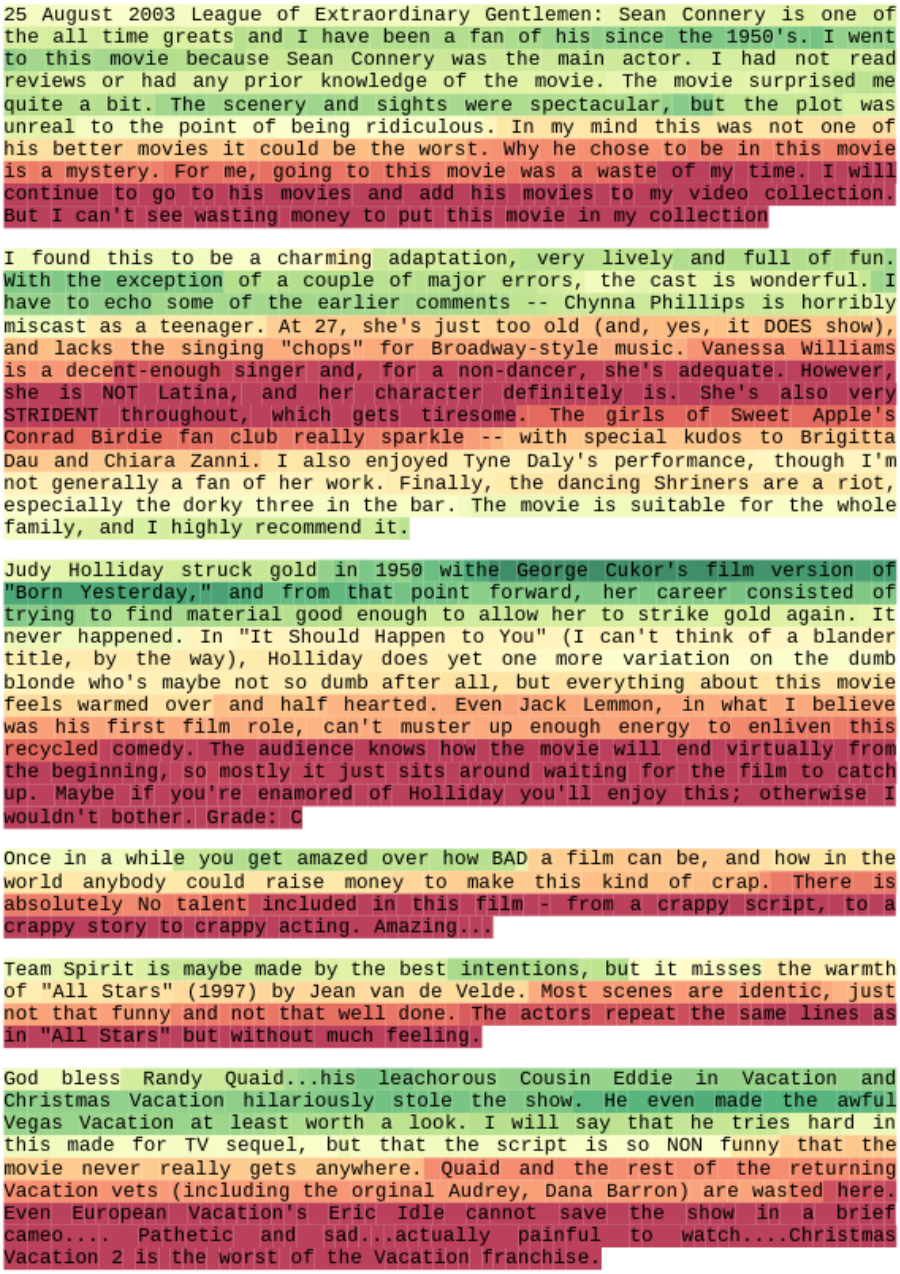

Figure 4.5: Visualizing the value of the sentiment cell as it processes six randomly selected high contrast IMDB reviews. Red indicates negative sentiment while green indicates positive sentiment. Best seen in color.[31]

Chapter 5

Datasets used in sentiment analysis

Sentiment analysis models require large labelled datasets to be highly effective. Since there are many different types of sentiment analysis, there is a need for a plethora of datasets. There are many datasets for a simple polarity classification, usually comprised of Twitter posts or reviews from platforms such as Amazon. However, it is a lot harder to obtain datasets for aspect-based sentiment analysis, even though, the platforms provide the functionality to rate different aspects, users are less likely to submit all of them. Furthermore, nearly all datasets are in English, which may create unintentional bias and cultural imbalance.

5.1 Pre-training dataset

This dataset was specifically collected for this thesis and contains posts from Twitter[1]. These posts were collected using Twitter streaming API during the three weeks and with the restriction that each text has to contain at least one Unicode emoji. Only the English tweets were collected.

All Twitter mentions, URLs and hashtags were removed during the cleaning process. Finally, all the leading and trailing whitespace characters were removed as well as all newline characters. The dataset contains 500 000 tweets split into training and validation sets with 75% for training and 25% for validation. An example from the dataset can be seen in Table 5.1.

Tweet
are you awake pal😂😂😂
i'm crying katy!😢😭
Mood 😑 All my local stores are like this
Sure is, we have cake twice in 5 days😍💕😘

Table 5.1: Example of tweets from the dataset.

An exploratory data analysis was performed to understand the structure of the tweets and usage of specific emojis. The tweet text length distribution is shown in Figure 5.1 where the box plot shows that median lenght is 40 characters, while 25% of tweets are under 25 characters and 75% of tweets are under 62 characters. The counts of the emojis indicate

[1] http://twitter.com/

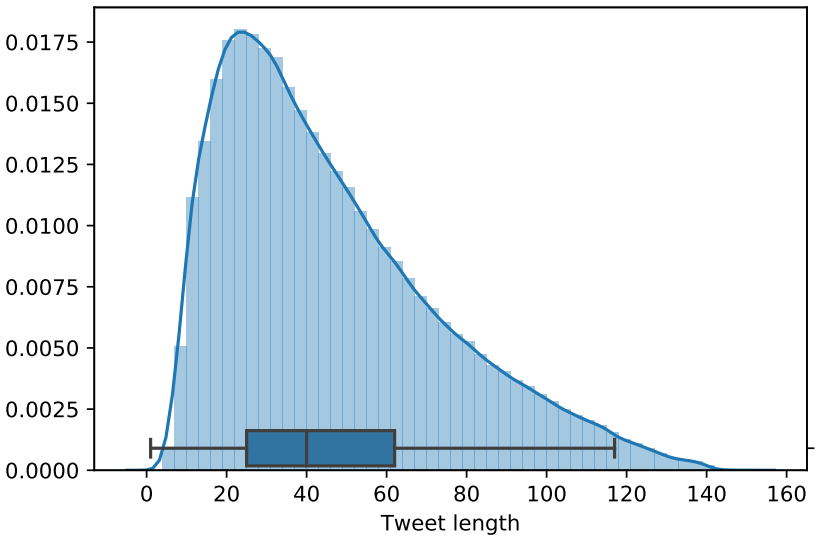

Figure 5.1: Tweet length distribution.

heavy use of two emojis where the first emoji is present in 36% of the tweets and the second in 21% of the tweets followed by the third with only 6% of the share. The top 10 emojis with their counts can be seen in Table 5.2.

Emoji	Occurrences
😂	179730
😭	106618
😍	33913
💜	23284
🔥	22920
🙏	19556
❤️	16700
💕	15439
😔	14563
😩	14222

Table 5.2: Top 10 emojis by occurrences in the dataset.

5.2 Benchmark datasets

The first dataset with identifier **SE0714** [39] consists of news headlines drawn from major newspapers such as the New York Times, CNN, and BBC News, as well as from the Google News search engine. Headlines are often used because of two main reasons. First, the news have typically a high load of emotional content, as they describe major national or worldwide events, and are written in a style meant to attract the attention of the readers. Second, the structure of headlines is suitable for the goal of conducting a sentence-level classification of emotions.

The class distribution is shown in Figure 5.2, where the fear represents almost 80% of the data samples. Examples from the dataset can be seen in Table 5.3.

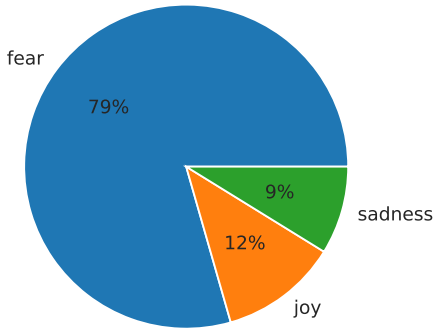

Figure 5.2: Class distribution from SE0714 dataset.

Class	Text
Fear	Mortar assault leaves at least 18 dead
Fear	Nasdaq fails in bid for LSE
Joy	Kate is marrying Doherty
Joy	World tourism sets record in 2006
Sadness	'House of Cards' actor Ian Richardson dead
Sadness	Parachutist dies at bridge-jump festival

Table 5.3: Examples from SE0714 dataset.

The inter-annotator agreement, described in Section 2.4, studies were conducted for each of the six emotions. The agreement evaluations were carried out using the Pearson correlation measure and are shown in Table 5.4

Emotion	Correlation
Fear	63.81
Joy	59.91
Sadness	68.19

Table 5.4: Inter-annotator agreement for each emotion in SE0714.[39]

The second dataset with identifier **Olympic** [37] contains tweets about London 2012 summer Olympics games. Social media platforms such as Twitter.com have become a common way for people to share opinions and emotions. Sports events are traditionally accompanied by strong emotions. Table 5.5 contains the statistics on inter-annotator agreement and the emotion agreement.

Emotion	Agreement
Polarity agreement	75.7
Emotion agreement	29.3

Table 5.5: Inter-annotator agreement in Olympic dataset.[37]

Dataset class distribution is shown in Figure 5.3, where the high control emotions represents 88% of the data samples. Control represents intensity. Examples from the dataset are shown in Table 5.6.

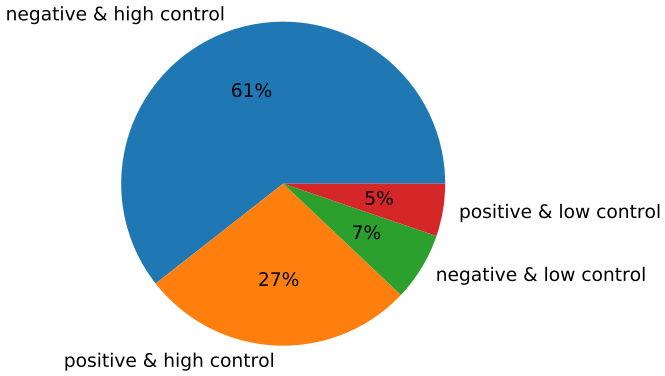

Figure 5.3: Class distribution from Olympic dataset.

Class	Text
Negative & high control	pierre yves beny is gonna be pissed about that stumble
Positive & high control	i want to do #dance and so bad laugh out loud #wish
Negative & low control	oh no the little ukrainian's face.. :(
Positive & low control	these white boy's so fucking fine, but they so short!

Table 5.6: Examples from Olympic dataset.

The third dataset identified as **PsychExp** [44] consists of self-reported emotional experiences created by a large group of psychologists. This dataset was constructed by a questionnaire given to each subject, where the objective was to recall occasions on which the subject experienced one of the following emotions: joy, fear, anger, sadness, disgust, shame or guilt.

Dataset class distribution is shown in Figure 5.4, where all classes are evenly distributed. Examples from the dataset are shown in Table 5.7.

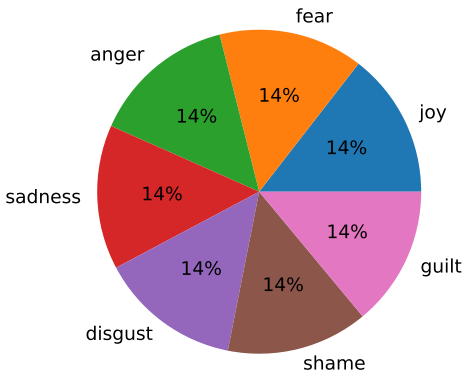

Figure 5.4: Class distribution from PsychExp dataset.

Class	Text
Joy	When I pass an examination which I did not think I did well.
Fear	At primary school the teacher caught me cheating during a dictation.
Anger	When a car is overtaking another and I am forced to drive off the road.
Sadness	When I lost the person who meant the most to me.
Disgust	When I found a bristle in the liver paste tube.
Shame	When one has been unjust, stupid towards someone else.
Guilt	When my uncle and my neighbour came home under police escort.

Table 5.7: Examples from PsychExp dataset.

There are two sentiment classification datasets from SentiStrength [41], **SS-Twitter** and **SS-Youtube**.

The **SS-Twitter** dataset contains tweets from public microblog broadcasts. The negative tweets are more prevalent in this dataset than positive tweets. Unusually for sentiment analysis, all the corpora are unbalanced, with highly unequal numbers of members of the different available categories.

Class distribution from the dataset is shown in Figure 5.5. There are almost 10% more negative classes than positive. Examples from the dataset are shown in Table 5.8.

Class	Text
Negative	Never by tea at Schiphol airport, it's expensive and you get a lousy tea!
Positive	I like my babe's tat there....

Table 5.8: Examples from SS-Twitter dataset.

The **SS-Youtube** dataset consists of text comments posted to videos on the Youtube website. This represents comments on resources and any associated discussions. There are

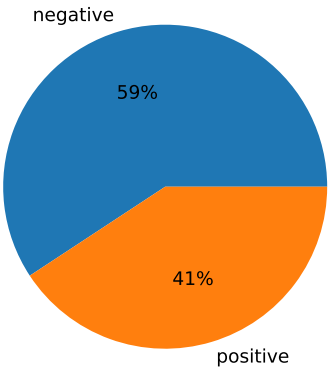

Figure 5.5: Class distribution from SS-Twitter dataset.

more negative data samples than positive, similarly to SS-Twitter dataset.

Class distribution from the dataset is shown in Figure 5.6, where almost 70% of data samples are negative. Examples from the dataset are shown in Table 5.9.

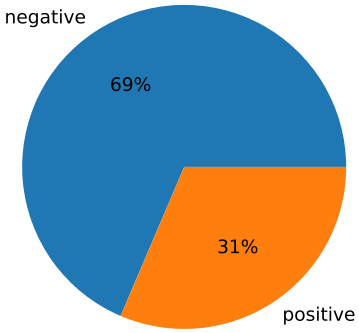

Figure 5.6: Class distribution from SS-Youtube dataset.

Class	Text
Negative	when the time comes for all to know it will be to late
Positive	Great video I love this song I heard it on OTH in an LP scene 3

Table 5.9: Examples from SS-Youtube dataset.

Finally, there are two datasets for sarcasm detection, **SCv1** and **SCv2-GEN**.

The **SCv1** dataset [43] contains posts extracted from the online debate site for political debate and discourse. The annotators were not given additional definitions of what it means for a post to be sarcastic. The inter-annotator agreement was 0.22 computed with Krippendorff's alpha as described in Section 2.4. The low agreement accords with native intuition – it is the class with the least dependence on lexicalization and the most subject to inter-speaker stylistic variation.

The classes are distributed evenly, as shown in Figure 5.7. Dataset examples can be seen in Table 5.10.

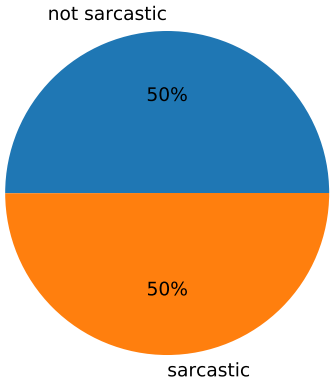

Figure 5.7: Class distribution from SCv1 dataset.

Class	Text
Not sarcastic	Please define „climb upward“ and „downhill slide“ in biologically meaningful terms. Please define „degeneration.“
Sarcastic	Old habits die hard and dogs don't pay attention to clocks!

Table 5.10: Examples from SCv1 dataset.

The **SCv2-GEN** dataset [25] is the second version of the previous dataset which aims to create more diverse and generic sarcasm dataset. In the task instructions, annotators were presented with a definition of sarcasm, followed by one example of a quote-response pair that contains sarcasm, and one pair that does not. This is a difference from the previous dataset, where the annotators were not given any additional information. The average per cent agreement with the majority vote was 89% for the three annotators of this dataset.

The classes are distributed evenly, as shown in Figure 5.8. Dataset examples can be seen in Table 5.11.

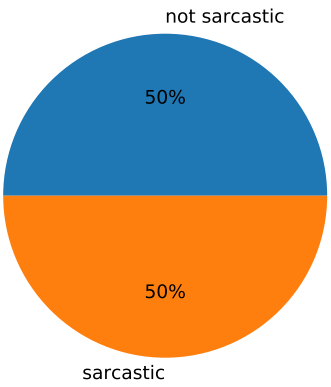

Figure 5.8: Class distribution from SCv2-GEN dataset.

Class	Text
Not sarcastic	If you feed me a lot of caffinated coffee i start to wiggle and squirm too.
	I'm not really in any pain.
Sarcastic	Yep, suppressing natural behavior is always the way to go.
	We should also get them to stop pooping.

Table 5.11: Examples from SCv2-GEN dataset.

Identifier	Task	Domain	Classes	Size
SE0714 [39]	Emotion	Headlines	3	1250
Olympic [37]	Emotion	Tweets	4	959
PsychExp [44]	Emotion	Experiences	7	7480
SS-Twitter [41]	Sentiment	Tweets	2	2113
SS-Youtube [41]	Sentiment	Video Comments	2	2142
SCv1 [43]	Sarcasm	Debate Forums	2	1995
SCv2-GEN [25]	Sarcasm	Debate Forums	2	3260

Table 5.12: Description of benchmark datasets.

Chapter 6

Proposed experiments

This chapter describes the proposed experiments of this thesis to explore and improve the non-supervised sentiment analysis methods described in Chapter 4. Firstly, the performance of the model mentioned in Section 4.3 indicates that language modelling task might lead to better representations across multiple tasks. Secondly, the distant supervision using emoticons causes models to learn good text representations indicated in Section 4.2. Furthermore, emoticons could be used to classify the emotional content of a text accurately. Finally, a few architectures are proposed based on these insights.

6.1 Emoji language modelling task

The proposed task is similar to the language modelling task, described in Section 3.5, where the probability is assigned to each sequence of words. Instead, it assigns a probability to the likelihood of a given emoticon, instead of a word, to follow the sequence which might contain both words and emoticons. Formally, the task is to estimate the probability of emoticon e following the given context c, which could be written as in Equation 6.1.

$$P(e_1, e_2, ..., e_n|c) = P(e_1|c)P(e_2|e_1, c)...P(e_n|e_1, e_2, ..., e_{n-1}, c) \tag{6.1}$$

$$\text{where } e \in E; E \subset V \tag{6.2}$$

The *context* is the sequence of words and emoticons from the vocabulary V, and the given emoticons E are a subset of the vocabulary.

6.2 Emoji model from scratch

This model has been inspired by sentiment neuron language model introduced in Section 4.3. However, instead of the mLSTM architecture mentioned in Section 3.4 uses the LSTM architecture described in Section 3.3. Similarly, the model processes input text as a sequence of characters including emoticons, meaning it is a character-level language model.

It is trained on the dataset created for purposes of this thesis by the author, described in Section 5.1. The input sequence characters and emoticons x are transformed into character-level embedding vectors x_c, described in Section 2.2.2. The embeddings are then passed into the LSTM unit. Afterwards, the encoded sequence represented by the output of the LSTM unit h is decoded by a linear layer defined in Equation 6.3.

$$y = Ah \tag{6.3}$$

Lastly, the decoded outputs y are masked and the loss is calculated only on outputs followed by emoticon, which is possible because the position of each emoticon in the input document is already known. The illustration of the architecture is shown in Figure 6.1. Each step shows most probable emoticon based on probability distribution from Softmax.

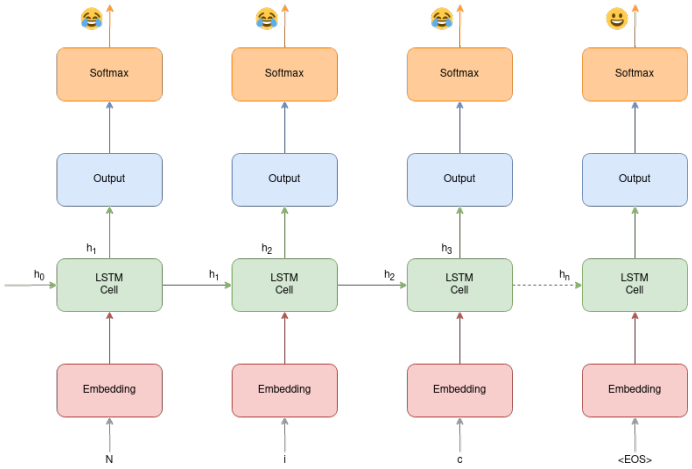

Figure 6.1: Emoji LSTM language model architecture.

6.3 Pre-trained Emoji model

Language model, as described in Section 3.5, training is expensive and requires large amounts of data. Expensive means that language model training is time-consuming and requires a lot of computing power.

The model builds upon the pre-trained GPT-2 architecture. It is trained on the dataset created for purposes of this thesis by the author, described in Section 5.1. The input sequence words and emoticons are encoded using Byte Pair Encoding, described in Section 3.6. The identifiers are transformed into embedding vectors and passed to the model, shown in Section 3.8. Afterwards, the encoded sequence represented by the output of the last layer of GPT-2 is decoded by linear layer, defined in Equation 6.3, which behaves as a language modelling layer. Lastly, the decoded outputs are masked and the loss is calculated only on outputs followed by an emoticon. The illustration of the architecture is shown in Figure 6.2.

6.4 Language model as a feature extractor

The experiment uses the language model after training as a feature extractor to explore sentiment captured in the text representations.

However, the representation vectors X passed into a model have to be pooled, considering the text representation always consists of a variable number of vectors N depending on the length of each text. Three different pooling strategies are used in this experiment.

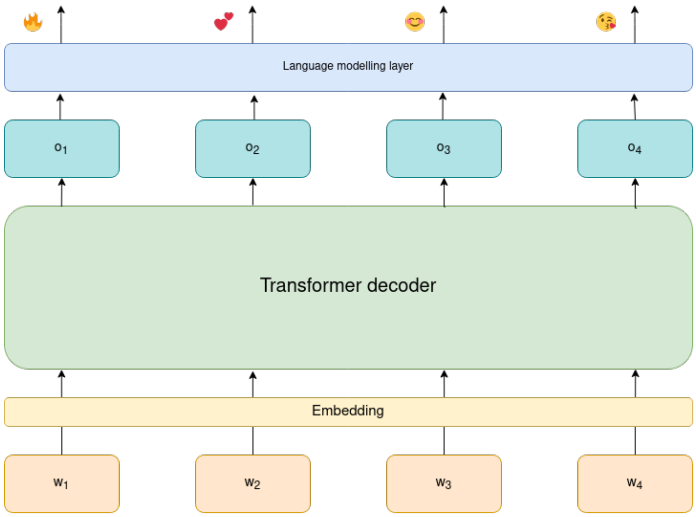

Figure 6.2: Emoji GPT-2 language model architecture.

- 'reduce mean' strategy takes the average of the prediction vectors on the time axis: $\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n$.
- 'reduce max' strategy takes the maximum of the prediction vectors on the time axis: $\bar{x} = max(X)$
- 'last token' strategy, which uses the prediction vector corresponding to the last token. $\bar{x} = x_n$

Finally, the pooled text representation vectors are used in the following experiments.

The first experiment is to calculate the correlation between the predicted emoticons and the target sentiment classes. Two different methods could be used to measure the association between two nominal variables: Cramér's V and Theil's U. The first variable is predicted emoticon classes, and the second is the target classes. These methods might indicate that there is a specific emoticon class with a high association. That would mean it could be used to predict a specific target class with high accuracy without any further supervision.

The second is to use predicted emoticons E to train the logistic regression model to predict the target sentiment classes Y. This method might find a combination of the emoticon predictions that predicts the target class y with high accuracy. That would mean high accuracy with small supervision.

And the last experiment is to compare Emoji GPT-2 model to original GPT-2 to see if there is any performance gain. Last hidden states H from both models are compared and the final emoticon predictions vector e with the hidden states h_m from both models. The goal of this experiment is to see if the outcome of the experiment from Section 6.3 provided more emotional content information to the GPT-2 model.

Chapter 7

Implementation and setup

This chapter presents the implementation and experimental setup used to train and evaluate experiments from Chapter 6. For the implementation of the models and experiments, Python 3.7 was used with PyTorch 1.4 framework. NVIDIA GeForce RTX 2080 Ti was used for training of the simple model from Section 6.2 and NVIDIA Tesla T4 was used for the larger GPT-2 model from Section 6.3.

7.1 Implementation

Specific architectures are in `model.py` file in each repository for that architecture. All experiments are in `notebooks` directory, and datasets used for those experiments are contained in the `data` directory.

Architecture from Section 6.2 was implemented only with PyTorch [27] framework, and for representations of emoticons is used library `emoji`. Dataset is loaded using `DataLoader` and as a sampler is used `RandomSampler`. Adam [18] is used as the optimizer, and the learning rate is adjusted during training using `linear_scheduler_with_warmup`. `CrossEntropyLoss` is utilised for calculating the loss during training. All metrics calculated during training such as training loss, evaluation loss and perplexity are logged using Tensorboard.

Architecture from Section 6.3 was implemented using PyTorch as well as `transformer` library. Also, for the representations of emoticons is used `emoji` library. Dataset is loaded the same way as described for previous architecture. Learning rate is adjusted using the same function as previous architecture. However, the optimizer used is AdamW [22], Adam algorithm with decoupled weight decay. `CrossEntropyLoss` is used in this case as well. Similarly, all metrics calculated during training are logged using Tensorboard.

7.2 Setup

Traning of architecture from Section 6.2 was done with learning rate 0.05 and clipping gradient norms at 1.5. The total number of epochs was 10, with warm-up steps set to 0. LSTM had 1 layer with 4096 unit hidden state and batch size was 256 during training. Dropout was set to 0.5.

Architecture from Section 6.3 was trained with the total number of epochs equal to 100, and warm-up steps set to 0 as well. Learning rate during training was 0.005, and weight decay for AdamW was set to 0. Maximum gradient normalization was set to 1.

Chapter 8

Results and Evaluation

The results of the experiments from the previous chapter are shown and discussed in this chapter. The models are evaluated on the test set of each benchmark dataset. The metrics used to evaluate the datasets are accuracy, for binary classification, and F1-score, for multiclass classification.

The results of the experiments are very surprising for many reasons. For example, the simple LSTM model was not able to learn any useful emoticon representations. However, Emoji GPT-2 model can achieve high results in sentiment, emotion and sarcasm classification.

8.1 Language model from scratch

The model in this experiment was not very successful. During pre-training, it started to diverge quickly on the validation set of the custom dataset described in Section. Furthermore, after inspecting the predictions on the validation set, it seemed to predict mostly two emoticons: 💯 and 💨. The example is shown in Figure 8.1.

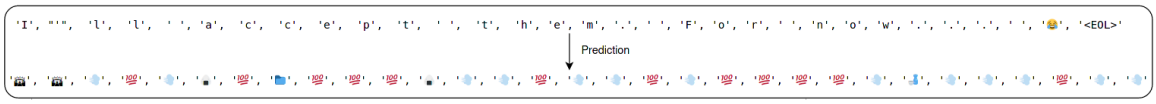

Figure 8.1: Emoji LSTM language model output example.

Predictions like these were not expected since occurrence counts of these emoticons are only 421 and 8186 respectively. Moreover, inspection by training a logistic regression on binary sentiment classification shown poor accuracy as well. Since the model did not perform well, it is not used in the following experiments.

Potential future work would be to find out the cause of the behaviour and improve the model architecture.

8.2 Pre-trained language model

The results of the model from this experiment are very interesting. The model with the best perplexity on the validation set is used in this experiment. The model exploration revealed that mostly the most common emoticons are predicted. A few random tweets

from the validation set were selected and used for the exploration. For each token, the top 5 most probable predictions were displayed. The example is shown in Figure 8.2.

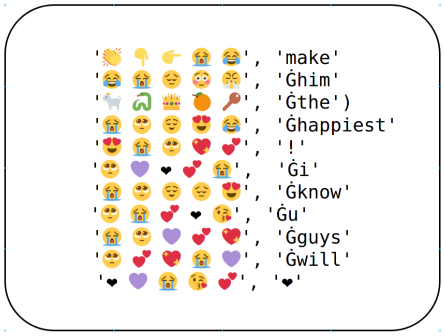

Figure 8.2: Emoji GPT-2 language model example.

The predictions in these results indicate that overall sentiment is encoded. For example, as shown in Figure 8.2, when the overall sentiment of the tweet is positive, mostly positive emoticons are predicted in the top 5. Also, the different emoticon predictions might indicate that different emotions might be encoded as well.

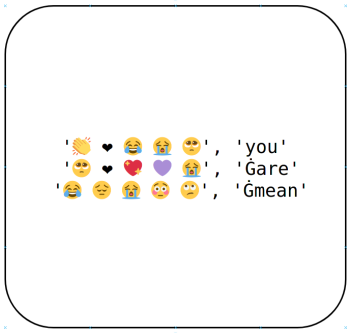

Figure 8.3: Emoji GPT-2 language model sarcasm example.

Furthermore, the example in Figure 8.3 might indicate that the emoticon predictions also encode sarcasm. Considering the word „mean“ in the context of the example is negative as suggested by 4 of the top 5 emoticons. However, the most probable prediction is still the positive emoticon, the face with tears of joy 😂.

Because the GPT-2 model is based on Transformer architecture, described in Section 3.7, it utilizes attention. The attention could be another way to gain more insights into the decision making of the model. Similarly to the previous case, a few random tweets were picked from the validation set and used to explore the attention heads of each layer.

Interestingly, the model seems to always assign high attention weights between the emoticon present in the tweet and the word it relates to emotionally. The examples are shown in Figure 8.4. Furthermore, this association seems to be always present in one of the first three heads of the seventh layer.

This insight could be further explored as a potential future work to see if it could be used for the non-supervised sentiment analysis.

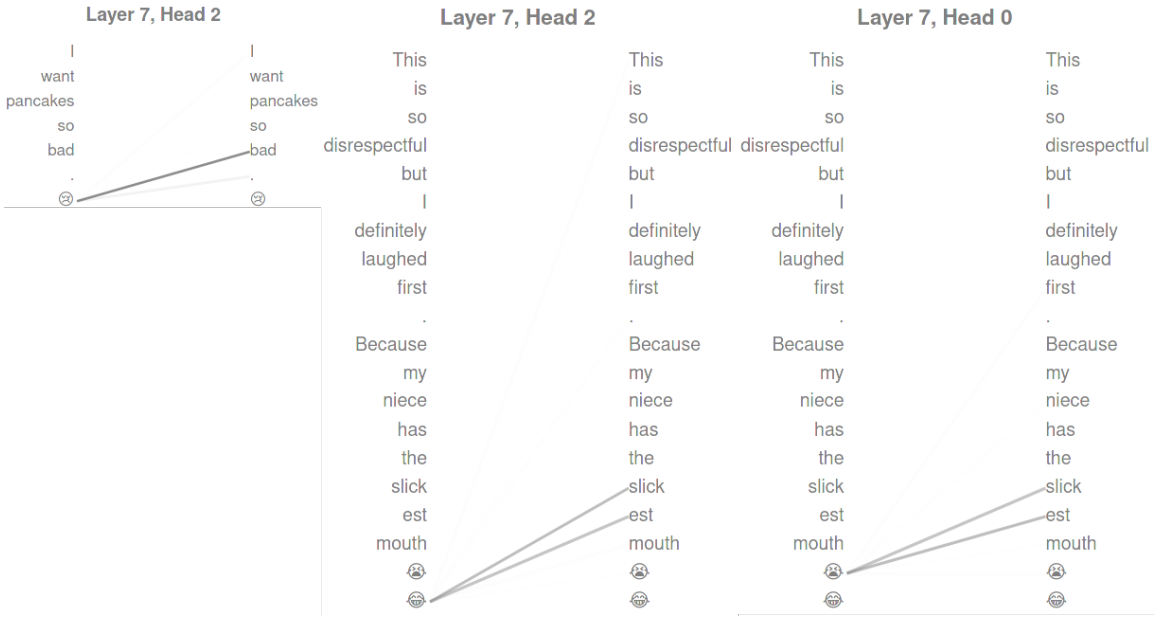

Figure 8.4: Emoji GPT-2 language model sarcasm example.

8.3 Measures of association

The measure of association between two variables refers to a method that measures the strength and direction of the relationship between those two variables. The Emoji GPT-2 model from the previous experiment is used in this experiment. Each text representation from this model was pooled using one of the strategies in Section 6.4. The final prediction was extracted from the pooled vector using $argmax$ function and used as an input to one of the methods used to calculate the measure of association. The second input variable is the target class. Both input text for the Emoji GPT-2 model and the target classes are from one of the benchmark datasets since this experiment was performed on all of them. Two methods were used to measure the similarity between these two discrete variables described in Section 3.9.

The results of the experiment, shown in Table 8.1, showed that the most predictive pooled vector was 'last token' on all benchmark datasets. It repeatedly showed significantly higher association values compared to the other two. Interestingly, the datasets for the sentiment classification task showed the highest degree of association, which might suggest that the overall sentiment of the text might be encoded better than for the other tasks. Moreover, the sarcasm datasets indicated the lowest degree of association confirming the difficulty of sarcasm classification.

Furthermore, the top 2 predictions were extracted from the pooled vector and used to calculate association. The results suggest the knowledge of all probable predictions increases the association, which means there is captured more information about the emotional content.

Finally, the difference between the results of both measures indicates that the Cramér's V might not be suitable for this experiment.

Identifier	Pooling strategy	Cramér's V	Theil's U
SE0714 [39]	reduce max	.0	.1424
	reduce mean	.1291	.0595
	last token	.0854	.0989
Olympic [37]	reduce max	.0	.1718
	reduce mean	.1166	.1196
	last token	.2016	.2179
PsychExp [44]	reduce max	.1981	.0635
	reduce mean	.1620	.0371
	last token	.2782	.1013
SS-Twitter [41]	reduce max	.3184	.1549
	reduce mean	.3266	.1287
	last token	.4028	.2016
SS-Youtube [41]	reduce max	.3936	.2009
	reduce mean	.4187	.2122
	last token	.4668	.2442
SCv1 [43]	reduce max	.0762	.0464
	reduce mean	.0519	.0211
	last token	.0657	.0281
SCv2-GEN [25]	reduce max	.1619	.0387
	reduce mean	.1374	.0206
	last token	.2048	.04581

Table 8.1: Measures of association on benchmark datasets.

8.4 Benchmark datasets

The experiment also uses the Emoji GPT-2 model from the experiment in Section 6.3. Also, the text representations from this model were pooled using each of the strategies in Section 6.4. These pooled text representations were used as features for the logistic regression model. The text inputs for the Emoji GPT-2 model and target classes used in the logistic regression model were from each of the benchmarks datasets.

The experiment benchmarks the model on three different tasks. There are three datasets for emotion classification, and two datasets for sentiment and sarcasm classification each. Each dataset was split into a training set and test set, the numbers of observations for each set are in Table 8.2. The logistic regression was trained using cross-validation with both L1 and L2 regularization. However, L2 regularized models always performed better.

Similarly to the previous experiment, the strategy which generated 'most predictive' pooled vectors was 'last token' on all the benchmark datasets. The best model with the highest accuracy of 89% on sentiment classification task was achieved on SS-Youtube dataset. This model was only 3% worse than the baseline model. Also, the model performed great on the sarcasm classification task where it achieved a comparable result with the baseline model on SCv2-GEN dataset. The biggest differences in performance compared to the baseline model were on the emotion classification datasets. Here, the model performed better on SE0714, where it scored 49% compared, which is 13% better than the baseline model. However, performed worse on the other two emotion classification datasets.

All results are shown in Table 8.2.

Identifier	Measure	N_{train}	N_{test}	Emoji GPT-2(last tok.)	DeepMoji(last)
SE0714 [39]	F1	250	1000	**.49**	.36
Olympic [37]	F1	250	709	.43	**.61**
PsychExp [44]	F1	1000	6480	.51	**.56**
SS-Twitter [41]	Accuracy	1000	1113	.83	**.87**
SS-Youtube [41]	Accuracy	1000	1142	.89	**.92**
SCv1 [43]	F1	1000	995	.64	**.68**
SCv2-GEN [25]	F1	1000	2260	.73	**.74**

Table 8.2: Comparison across benchmark datasets.

The last experiment with Emoji GPT-2 showed different all three classification task datasets. It performed better on Olympic dataset than the original GPT-2, however, a little worse on the other two emotion task datasets. Overall, the last emoticon predictions seem to be a better representation than the hidden state of the Emoji GPT-2.

Furthermore, the Emoji GPT-2 performed better on the sentiment task datasets where it achieved better results compared to both hidden state representations of Emoji GPT-2 as well as hidden representations of the original GPT-2. Again, the last emoticon predictions seem to be in this case better representation, than both Emoji GPT-2 and GPT-2 hidden states.

Lastly, Emoji GPT-2 indicated worse performance compared to original GPT-2. The reason why there is a difference in the performance on sarcasm task could be a part of future research.

It should be noted that all bechmarks with better Emoji GPT-2 are from social media domain as well as the pre-training dataset and might be reason for better performance. For results see Table 8.3.

Identifier Pooling	Emoji GPT-2 last token probabilities	Emoji GPT-2 last hidden	GPT-2 last hidden
SE0714 [39]	.49	.49	**.53**
Olympic [37]	**.43**	.42	.32
PsychExp [44]	.51	.51	**.53**
SS-Twitter [41]	**.83**	.82	.77
SS-Youtube [41]	**.89**	.87	.84
SCv1 [43]	.64	.65	**.67**
SCv2-GEN [25]	.73	.73	**.78**

Table 8.3: Comparison with original GPT-2 across benchmark datasets.

8.5 Summary

The experiments described in previous sections rendered the following conclusions:

- The simple LSTM architecture does not seem to be able to learn any useful text representations during pre-training on introduced emoticon language modelling task. However, the results from the Emoji GPT-2 show that the model can learn emotional content, so further exploration of the capabilities of the LSTM architecture is left as a potential future work.

- Pre-training on the emoticon language modelling task seems 'to teach the emotional content knowledge to the model' suggested by context-dependent emoticon predictions. Also, the emotional content knowledge was present during the analysis of attention heads of the model and the further inspection of this phenomenon is left as future work.
- The model emoticon predictions seem to encode emotional content information based on the results of the model compared to the baseline model and the original GPT-2 model. The results suggest that the overall sentiment of the text is the most captured phenomena out of all three benchmark tasks. Furthermore, the results show that the emoticon language modelling task is successful in encoding more emotional content in the model predictions compared to original GPT-2 model.

Chapter 9

Conclusion

This thesis discussed the phenomenon of sentiment in natural language and currently available methods for non-supervised sentiment analysis. Custom pre-training dataset was collected from Twitter, and various benchmark datasets for sentiment classification, emotion detection and sarcasm classification were introduced. Experimental training task was created based on the observations from two current methods. The first observation is that language modelling task can capture sentiment from the text. The second observation is that emoticons can be used as a form of distant supervision with state-of-the-art results as shown in [14], and described in Section 4.2. And finally, different experiments with the experimental training task were introduced, and the results were compared against the baseline architecture. Ablation study was not done because no suitable ablation was found.

The baseline architecture was DeepMoji architecture (see Section 4.2), where all layers were frozen, and only the last layer was fine-tuned on the downstream dataset. The whole architecture was pre-trained on Twitter data with emoticons as supervision. Emoticons present in the data sample were used as the annotation separately for the whole sample. Meaning that if there were three different emoticons, the data sample would be present in the training dataset three times with different annotation emoticon. Duplicate emoticons were thrown away. DeepMoji is bidirectional discriminative architecture. Contrary to DeepMoji, Emoji GPT-2 did not remove duplicate emoticons, it is generative architecture which models sequences of emoticons and it is single directional.

It is clear from the results of the experiments that overall sentiment, emotions and even sarcasm can be extracted from the text using the experimental task. The simple LSTM architecture was not able to learn meaningful representations. Further exploration of the cause was left as a potential future work.

However, extended GPT-2 architecture showed comparable results with the baseline architecture as well as performance improvements over the original GPT-2 architecture. The Emoji GPT-2 architecture outperformed the baseline DeepMoji architecture only in one of the emotion classification benchmarks. Nevertheless, the results on the sentiment classification and sarcasm classification tasks were comparable. Furthermore, since the Emoji GPT-2 uses pre-trained GPT-2, it required less training time and data to achieve similar results.

The use of Transformer based architecture (GPT-2) opens a possibility to perform analysis on the attention heads. Interesting attention groups were discovered during the attention analysis in the Emoji GPT-2. It would be interesting to see if it would be possible to classify sentiment only based on the attention heads. However, further research in this area was left as a potential future work.

Conclusion based on the results of the experiments done in this thesis confirms that language modelling task is capable of extracting sentiment from the text. Also, the proposed experiment emoji modelling task can capture better emotional representations than the hidden Transformer representation by almost 5 per cent in some benchmark tasks. Moreover, emoticons as a form of supervision are great at capturing overall sentiment, different emotions as well as sarcasm. Potential future improvements in this area would be to explore LSTM architecture more in-depth. Moreover, explore the potential of attention based classification. Other future path of research could be fine-tuning the whole model on the downstream datasets.

Bibliography

[1] Ba, J. L., Kiros, J. R. and Hinton, G. E. *Layer Normalization*. 2016.

[2] Bengio, Y., Ducharme, R., Vincent, P. and Janvin, C. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.* JMLR.org. march 2003, vol. 3, null, p. 1137–1155. ISSN 1532-4435.

[3] Bergsma, W. A bias-correction for Cramér's V and Tschuprow's T. *Journal of the Korean Statistical Society*. 2013, vol. 42, no. 3, p. 323 – 328. Available at: http://www.sciencedirect.com/science/article/pii/S1226319212001032. ISSN 1226-3192.

[4] Bishop, C. M. *Pattern recognition and machine learning*. Springer, 2006.

[5] Cambria, E., Schuller, B., Xia, Y. and Havasi, C. New Avenues in Opinion Mining and Sentiment Analysis. *IEEE Intelligent Systems*. 2013, vol. 28, no. 2, p. 15–21.

[6] Cambria, E., Song, Y., Wang, H. and Howard, N. Semantic Multidimensional Scaling for Open-Domain Sentiment Analysis. *IEEE Intelligent Systems*. 2014, vol. 29, no. 2, p. 44–51.

[7] Capital, I. *Brexit hourly sentiment data* [online]. June 2016 [cit. 2020-05-28]. Available at: https://twitter.com/IkoveCapital/status/746036922024329217/photo/1.

[8] Chevalier, G. *Long short-term memory* [online]. [cit. 2020-05-27]. Available at: https://en.wikipedia.org/wiki/Long_short-term_memory.

[9] Chou, S.-Y. *AUC - Insider's Guide to the Theory and Applications* [online]. [cit. 2020-05-27]. Available at: https://sinyi-chou.github.io/classification-auc/.

[10] CRAMÉR, H. *Mathematical Methods of Statistics (PMS-9)*. Princeton University Press, 1999. Available at: http://www.jstor.org/stable/j.ctt1bpm9r4. ISBN 9780691005478.

[11] De Clercq, O. The many aspects of fine-grained sentiment analysis: An overview of the task and its main challenges. In: IARIA. *HUSO 2016*. 2016, p. 23–28.

[12] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018.

[13] Ekman, P. An argument for basic emotions. In:. 1992.

[14] Felbo, B., Mislove, A., Søgaard, A., Rahwan, I. and Lehmann, S. *Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm*. 2017.

[15] Goldberg, Y. *Neural Network Methods in Natural Language Processing*. 2017.

[16] Hochreiter, S. and Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* Cambridge, MA, USA: MIT Press. november 1997, vol. 9, no. 8, p. 1735–1780. Available at: https://doi.org/10.1162/neco.1997.9.8.1735. ISSN 0899-7667.

[17] Howard, J. and Ruder, S. *Universal Language Model Fine-tuning for Text Classification*. 2018.

[18] Kingma, D. P. and Ba, J. *Adam: A Method for Stochastic Optimization*. 2014.

[19] Krause, B., Lu, L., Murray, I. and Renals, S. *Multiplicative LSTM for sequence modelling*. 2016.

[20] Krippendorff, K. Reliability in Content Analysis: Some Common Misconceptions and Recommendations. *Human Communication Research*. january 2006, vol. 30, no. 3, p. 411–433. Available at: https://doi.org/10.1111/j.1468-2958.2004.tb00738.x. ISSN 0360-3989.

[21] Lange, R. T. Inter-rater Reliability. In: Kreutzer, J. S., DeLuca, J. and Caplan, B., ed. *Encyclopedia of Clinical Neuropsychology*. New York, NY: Springer New York, 2011, p. 1348–1348. Available at: https://doi.org/10.1007/978-0-387-79948-3_1203. ISBN 978-0-387-79948-3.

[22] Loshchilov, I. and Hutter, F. *Decoupled Weight Decay Regularization*. 2017.

[23] Mikolov, T., Chen, K., Corrado, G. and Dean, J. *Efficient Estimation of Word Representations in Vector Space*. 2013.

[24] Nair, V. and Hinton, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Madison, WI, USA: Omnipress, 2010, p. 807–814. ICML'10. ISBN 9781605589077.

[25] Oraby, S., Harrison, V., Reed, L., Hernandez, E., Riloff, E. et al. Creating and Characterizing a Diverse Corpus of Sarcasm in Dialogue. In: *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Los Angeles: Association for Computational Linguistics, September 2016, p. 31–41. Available at: https://www.aclweb.org/anthology/W16-3604.

[26] Pascanu, R., Mikolov, T. and Bengio, Y. *On the difficulty of training Recurrent Neural Networks*. 2012.

[27] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Wallach, H., Larochelle, H., Beygelzimer, A., Alché Buc, F. d, Fox, E. et al., ed. *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, p. 8024–8035. Available at: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[28] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C. et al. *Deep contextualized word representations*. 2018.

[29] Plutchik, R. Emotions: A general psychoevolutionary theory. *Approaches to emotion*. 1984, vol. 1984, p. 197–219.

[30] Radford, A. Improving Language Understanding by Generative Pre-Training. In:. 2018.

[31] Radford, A., Jozefowicz, R. and Sutskever, I. *Learning to Generate Reviews and Discovering Sentiment*. 2017.

[32] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. et al. Language Models are Unsupervised Multitask Learners. In:. 2019.

[33] Saif, H., Fernandez, M. and Alani, H. Evaluation Datasets for Twitter Sentiment Analysis. A survey and a new dataset, the STS-Gold. In:. December 2013.

[34] Sennrich, R., Haddow, B. and Birch, A. Neural Machine Translation of Rare Words with Subword Units. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, August 2016, p. 1715–1725. Available at: https://www.aclweb.org/anthology/P16-1162.

[35] *Sentiment definition*. Available at: https://dictionary.cambridge.org/dictionary/english/sentiment.

[36] Seyeditabari, A., Tabari, N. and Zadrozny, W. *Emotion Detection in Text: a Review*. 2018.

[37] Sintsova, V., Musat, C. and Pu, P. Fine-Grained Emotion Recognition in Olympic Tweets Based on Human Computation. In: *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Atlanta, Georgia: Association for Computational Linguistics, June 2013, p. 12–20. Available at: https://www.aclweb.org/anthology/W13-1603.

[38] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D. et al. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, October 2013, p. 1631–1642. Available at: https://www.aclweb.org/anthology/D13-1170.

[39] Strapparava, C. and Mihalcea, R. SemEval-2007 Task 14: Affective Text. In: *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, p. 70–74. Available at: https://www.aclweb.org/anthology/S07-1013.

[40] Su, F. and Markert, K. From Words to Senses: A Case Study of Subjectivity Recognition. In: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Manchester, UK: Coling 2008 Organizing Committee, August 2008, p. 825–832. Available at: https://www.aclweb.org/anthology/C08-1104.

[41] Thelwall, M., Buckley, K. and Paltoglou, G. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*. 2012, vol. 63, no. 1, p. 163–173. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.21662.

[42] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al. *Attention Is All You Need*. 2017.

[43] Walker, M., Tree, J. F., Anand, P., Abbott, R. and King, J. A Corpus for Research on Deliberation and Debate. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), May 2012, p. 812–817. Available at: http://www.lrec-conf.org/proceedings/lrec2012/pdf/1078_Paper.pdf.

[44] Wallbott, H. G. and Scherer, K. R. How universal and specific is emotional experience? Evidence from 27 countries on five continents:. In:. 1986.

[45] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. et al. *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. 2019.

[46] Zeng, Z., Zhou, W., Liu, X. and Song, Y. *A Variational Approach to Weakly Supervised Document-Level Multi-Aspect Sentiment Classification*. 2019.