

Quantifying Emotional Valence, Arousal and Dominance from Natural Language

Ana Sofia Aparício da Costa

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Doutor Bruno Emanuel da Graça Martins

Examination Committee

Chairperson: Prof. Doutor Alberto Manuel Rodrigues da Silva

Supervisor: Prof. Doutor Bruno Emanuel da Graça Martins

Members of the Committee: Prof. Doutor Ricardo Daniel Santos Faro Marques Ribeiro

September 2020

Acknowledgements

I would like to express my deep gratitude to Professor Doutor Bruno Martins, my supervisor, for his guidance, enthusiastic encouragement and his availability for the countless meetings through this research journey.

I would also like to thank the principal investigator of the Project HATE, Professor Susana Salgado, financed by Fundação para a Ciência e a Tecnologia, through the grant with reference PTDC/CPO-CPO/28495/2017.

A special thank you to my parents, brother and grandparents for all the love and support through all these years. Thank you for your guidance, support and hard work, that enabled me to be the person I am today.

This journey would not be possible without the support of my family and friends, for sharing so many wonderful moments and making me want to be a better person each day.

Finally, to my boyfriend, for all the love, support and patience through the last year.

For my family,

Abstract

The growth of social media platforms has drawn attention to natural language processing, especially to the sentiment analysis field. Previous studies have covered several classification and regression methods to quantify emotions expressed in textual documents, in particular, supervised models leveraging hand-labelled training datasets. Many of these studies have also relied on neural network methods. However, to the best of my knowledge, there is still a gap when using deep learning to infer emotions from languages that have few or no training resources. This M.Sc. thesis compares the use of multi-layer perceptrons (MLP), recurrent neural networks based on long short-term memory units (LSTM), convolutional neural networks (CNN), and different types of attention mechanisms, as well as cross-language embeddings, as an approach to combine training data in multiple languages to extend the state-of-the-art in this field of emotion analysis. The proposed methods were evaluated with datasets used in previous studies, annotated with ratings regarding valence, arousal and dominance, in several languages. The obtained results support the understanding that machine learning (ML) models can predict emotions expressed in text, even in several languages. The proposed methods perform comparably, and even outperform, previous work in this field, that mostly produced models that use only monolingual data.

Keywords

Natural Language Processing; Sentiment Analysis; Neural Networks; Emotional Ratings; Multilingual Analysis

Resumo

O crescimento das redes sociais chama atenção sobre a área do processamento da língua natural, em particular sobre a área de análise de sentimentos. Estudos anteriores utilizam vários métodos de classificação para a quantificação de emoções em textos, em particular recorrendo a, modelos supervisionados que utilizam conjuntos de dados previamente anotados. Muitos destes estudos também dependem de métodos como redes neurais profundas. No entanto, tanto quanto se sabe, ainda há uma carência quando se trata da utilização de redes neurais profundas para inferir emoções de línguas que tenham poucos dados de treino disponíveis. Esta tese avalia comparativamente várias técnicas para a quantificação de emoções, nomeadamente perceptrões multicamada (MLP), redes recorrentes (LSTM), redes convolucionais (CNN) e mecanismos de atenção, bem como embeddings trans-linguísticos com o objetivo de combinar dados de múltiplas línguas, com vista a estender o estado da arte nesta área. Os métodos propostos foram validados com datasets usados em estudos anteriores, considerando valência, entusiasmo e dominância em várias línguas. Os resultados obtidos suportam a afirmação que modelos de aprendizagem conseguem prever emoções expressas textualmente, mesmo em várias línguas. Foram obtidos resultados comparáveis, e até superiores, a trabalhos anteriores neste ramo, mesmo comparando com modelos que preveem emoções para dados monolíngues.

Palavras-Chave

Processamento de Linguagem Natural; Análise de Sentimento; Redes Neurais; Valor Emocional; Análise Multilíngue

Contents

1	Introduction	3
1.1	Motivation	3
1.2	Research Objectives	4
1.3	Methodology	5
1.4	Results	6
1.5	Contributions	6
1.6	Master Thesis Structure	7
2	Fundamental Concepts and Related Work	9
2.1	Fundamental Concepts	9
2.1.1	Encoding Textual Information	9
2.1.2	Introduction to Neural Networks and Deep Learning	12
2.1.3	Emotion Representation	20
2.2	Related Work	21
2.2.1	Assigning Sentiment to Words	22
2.2.2	Assigning Emotion to Textual Utterances	27
2.3	Overview	32
3	Quantify Emotion in Multiple Languages	33
3.1	Text Representation in a Multilingual Space	33
3.2	Proposed Models	35
3.2.1	Quantify Emotion from Words	36
3.2.2	Quantify Emotion from Textual Utterances	39

3.2.2.1	Simple Models Exploring Averages	39
3.2.2.2	Models Exploring Machine Learning	40
3.3	Overview	44
4	Experimental Evaluation	45
4.1	Datasets	45
4.1.1	Word Affective Norms	45
4.1.2	Text Affective Norms	47
4.2	Evaluation Metrics	47
4.3	Experimental Results	48
4.3.1	Assigning Affective Norms to Words	49
4.3.2	Assigning Affective Norms to Textual Utterances	53
4.3.2.1	Simple Models Exploring Averages	54
4.3.2.2	Models Exploring Machine Learning	54
4.4	Overview	57
5	Conclusions and Future Work	59
5.1	Main Results	59
5.2	Future Work	60
	Bibliography	71

List of Figures

2.1	Dense word representations in a vector space.	10
2.2	FastText	11
2.3	Multi-Layer Perceptron	13
2.4	Convolution and polling applied to the sentence <i>the dog runs after the cat.</i>	16
2.5	Summarizing the behaviour of a RNN.	17
2.6	Summarizing the behaviour of a LSTM.	18
2.7	Summarizing the behaviour of a GRU.	19
2.8	Self-Assessment Manikin	21
2.9	Valence, arousal and dominance three-dimensional space	21
2.10	Method proposed by Calvo and Mac Kim (2013).	28
2.11	Sequence convolution neural networks unification by concatenation.	29
2.12	CNN and LSTM model proposed by Köper et al. (2017)	30
2.13	Pipeline for inferring emotion based on studies conducted by Kratzwald et al. (2018).	31
3.1	Cross-lingual alignment method from Conneau et al. (2017).	34
3.2	Models for Assigning Word-level Sentiment	40
3.3	Convolution and Polling operations applied to a sentence.	41
3.4	A model based on a BiLSTM and attention.	42
3.5	Proposed models applying Self-Attention and BiLSTM Layers.	43
4.1	Comparison of the dimensional distribution of the words of the datasets in several languages.	46

4.2	Comparison of the dimensional distribution of the datasets in several languages.	48
4.3	The absolute error between the affective norms predicted by the different models and the expected results.	52
4.4	Example of cross-validation with multiple datasets.	55
4.5	Website design in a prototype for predicting sentiment associated to input sentences.	58

List of Tables

2.1	Most important studies related to assigning sentiment to words.	27
2.2	Most important works in Assigning sentiment to Textual Utterances	32
4.1	Obtained results when predicting ratings for words in the English ANEW, War-riner and Glasgow lexicons. The associated p -values for the Pearson product-moment correlation coefficient were always lower than 0.001.	49
4.2	Pearson correlations obtained when predicting the ratings in four different adap-tations of the ANEW lexicon, namely for the Spanish, Portuguese, Italian and German languages. The corresponding p -values were always lower than 0.001. . .	50
4.3	Correlations between human norms for English words and human norms in the four different adaptations of the ANEW lexicon. The corresponding p -values were always lower than 0.001.	51
4.4	Obtained results, in terms of Pearson's correlation coefficient, when using mono-lingual data through a leave-one-out cross validation methodology. The corre-sponding p -values were always below 0.001.	53
4.5	Results obtained for statistical sentiment prediction of textual utterances, in terms of Pearson's correlation coefficient and Mean Absolute Error (MAE). . . .	54
4.6	The prediction of valence, arousal and dominance with several models. The train-ing and testing data are textual utterances form datasets in English, Polish and Portuguese.	55

Acronyms

BE Basic Emotions

CART Classification and Regression Trees

CBOW Continuous Bag of Words

CNN Convolutional Neural Network

GRU Gated Recurrent Unit

HAL Hyperspace Analogue to Language

kNN k -nearest neighbour

LSA Latent Semantic Analysis

LSTM Long Short-Term Memory

MAE Mean Absolute Error

MSE Mean Squared Error

ML Machine Learning

MLP Multi-Layer Perceptron

NLP Natural Language Processing

NN Neural Network

OOV Out-Of-Vocabulary

ReLU Rectified Linear Unit

RNN Recurrent Neural Network

SAM Self-Assessment Manikin

SCNN Sequence-based Convolutional Neural Network

SGD Stochastic Gradient Descent

S-RNN Simple Recurrent Neural Network

SVD Singular Value Decomposition

TASA Touchstone Applied Science Associates

UMWE Unsupervised Multilingual Word Embeddings

1 Introduction

Sentiment analysis has been one of the main application areas for Natural Language Processing (NLP) leveraging neural networks, aiming at the extraction of either negative and positive evaluations, or estimating emotions. Human emotional ratings are frequently used within cognitive science, behavioural psychology and psycholinguistic research (Perugini and Bagozzi, 2001), social media analysis (Hutto and Gilbert, 2014), among others, motivating the interest on automated methods for quantifying opinions expressed in textual documents.

This chapter is divided in six sections. Section 1.1 briefly summarizes the related work in this field, discussing what is not covered in this particular area and stating the research question. Section 1.2 puts forward the research objectives that will allow answering the main question, while Section 1.3 presents the methodology, outlining the methods used to answer the question. At the end of this section, there is an overview of the results and main contributions of this work. The chapter ends with an overview of the structure of this dissertation.

1.1 Motivation

Previous studies have covered the polarity (positive vs negative) of subjects discussed in textual documents (Wilson et al., 2005), as well as the quantification of emotions expressed in text. Two main families of methods have been developed to represent human emotions (Ekman and Friesen, 1971). One is categorical, based on six universal Basic Emotions (BE) (Ekman, 1992). The other is dimensional, advocating continuous numerical values that progress through multiple dimensions (Wang et al., 2016). Studies have showed that the categorical method does not consider to be opposite references on the dimension representation.

Since it takes a significant amount of human resources to annotate words and textual utterances regarding sentiment and/or emotions, there is significant interest in producing automatic methods. In previous studies, to infer the emotion of words in lexicons, researchers have used techniques based on word co-occurrence patterns leveraging Latent Semantic Analysis (Bestgen and Vincze, 2012), Point-wise Mutual Information statistics, and/or words in regression

modelling approaches (Mandera et al., 2015). Buechel and Hahn (2018) developed a multi-task learning based on neural networks for predicting the emotion of words in three dimensions.

When considering complex syntactical structures, such as sentences and large pieces of text, it is necessary to consider more complex models. Binali et al. (2010) recognised three different approaches for emotion detection: keyword-based, learning-based and hybrid-based (a fusion between the other two). The three methods rely on different linguistic analysis tools.

A simple approach to infer sentiment on text was developed by Ma et al. (2005). It used keywords spotting applied to a chat system in order to generate emotionally responsive messages. Malheiro et al. (2016) used a keyword approach to analyse song verses, considering the valence and arousal space. However, word-level emotion prediction has some limitations.

Calvo and Mac Kim (2013) developed a model, using TF-IDF representations, to infer the most relevant words when classifying a text. A few years later, three mechanisms (i.e. linear regression, a Multi-Layer Perceptron (MLP) model, and a model composed of two stacked Long Short-Term Memory (LSTM) units) were used to classify text according (Köper et al., 2017) to four emotions: anger, fear, joy, and sadness. Zahiri and Choi (2017) conducted emotion detection of the TV show Friends, through a Sequence-based Convolutional Neural Network (SCNN).

These previous methods were mainly applied to text written in the English language. To the best of my knowledge, there is still a gap when using deep learning and neural networks to quantify sentiment and emotional dimensions in text from languages with few or none training resources. My M.Sc. research project explored alternatives for cross-lingual analysis of emotions expressed in textual contents.

1.2 Research Objectives

To address the main research objective, it was necessary to understand how to infer emotion ratings first from words, and after from larger textual utterances. This separation allowed me to formulate secondary questions regarding each separate experiment.

When considering words, it was first crucial to understand how well a Machine Learning (ML) model can infer emotion rating in a monolingual scenario. Following this question, it was necessary to know how well a ML model, trained with English lexicons, can predict emotion on other languages, how the correlation between the English norms and those for the other languages can affect the results, and what model would perform better.

When considering large textual utterances, it would be fascinating to understand if there is a difference between models that predict word-level emotion rating, or text-level. Through experiments, I also tried to understand what type of model will perform better, namely a Convolutional Neural Network (CNN)'s or an LSTM's, and in some cases understanding if there is benefits of including a pre-trained components.

1.3 Methodology

For the text to be provided as input to ML models, it is first necessary to convert the text into a numeric representation that the model will understand. There are some techniques to map words into vectors of numbers. The one used in this work is called FastText (Grave et al., 2018). This method was chosen because of its ability to generate embedding even for words that were never seen in the data used to learn the representations (i.e., uncommon words, spelling mistakes). In this thesis, the framework named UMWE from Chen and Cardie (2018) was also used to convert several language embeddings into one target embedding space through a translation matrix. In this study, the target language was English.

For the word experiments, it was necessary to train models (i.e. k -nearest neighbour (kNN), regression, random forest, kernel ridge regression and MLPs) with the emotion ratings in the lexicons from Warriner et al. (2013) and from Scott et al. (2019), including words that did not appear in the ANEW lexicon from Bradley and Lang (1999) corpus. The models were then tested using the words that appear on ANEW.

For the experiments regarding sentences, I first trained an MLP with emotion ratings in lexicons from six different languages. Several other models were also produced to understand if it was necessary to access the entire syntactic structure to determine emotion ratings form a text. Four models that do not take into consideration the syntactic structure and do not require training were created. For instance an *MLP + Average* model, using the pre-trained MLP, calculates an average of the sentiment prediction of all the words to show the sentiment of the sentence. An *Average + MLP* model is similar to the previous one, but instead it calculates a mean of the word embedding and then uses the MLP used the embeddings average to make a prediction. The last two models are more complex. The first considers windows of sizes between one and five words. Then, the average of all these pooling windows was calculated before applying the MLP. The last model suffered a little change since the MLP was applied after each pooling window, and I then calculated the average of the ratings.

Still on what regards experiments with sentences, eight trainable models were conceived and validated with two-fold cross-validation (i.e. LSTM, MLP+LSTM, CNN, MLP+CNN, CNN+MLP, Attention Concat, Attention Feacture Bassed, Attention Affine Transformation). The last three models were based on proposals from Margatina et al. (2019), and they also use word-level predictions in the definition of neural attention mechanisms.

1.4 Results

The results of the words experiments are promising, especially with the kernel ridge and MLP models. They show that a ML model can predict outcomes comparably to human annotators with both monolingual and multilingual data. The results also show that relatively high correlations can be achieved for all five languages. However, the cross-lingual results are inferior to the results obtained for the monolingual setting. It was also interesting to notice that higher predictive accuracy is generally also obtained for languages where the correlation towards the English norms is higher (i.e. Italian and Spanish).

In turn, the results for the sentence level experiments show that three trained models generally performed better (Attention Concat, Attention Feacture Bassed, Attention Affine Transformation). However, the average word-level prediction model also showed promising results. LSTM models tend to perform slightly better than CNN models, and the difference was more evident in the arousal dimension. However, when the CNN and LSTM models were aligned with the pre-trained MLP, the results decreased, showing that a combination with a pre-trained MLP can even decrease the performance of the model, if not designed carefully.

1.5 Contributions

The obtained results support the understanding that a ML model can predict emotion ratings, even in several languages. The main contribution of this work relies on the amount of models validated to infer how to extract emotions from both words and large textual utterances. There are few works on emotions quantification, in particular considering cross-lingual settings or the dimensional way of quantifying sentiment. This thesis provides three trained models and one word-level model, all showing promising results compared to the state-of-the-art.

Alongside this, I also created a website to showcase the results, and through which the users can predict the sentiment of a textual utterance. In this website, it is possible to insert a

sentence in a given language. When the model provides a prediction, the site shows the results aligned with the relevance of each word.

Implementations for the models that support the word-level and sentence-level representations reported on this dissertation are now publicly available in a GitHub repositories. I did one repository that contains the word-level experiments¹, one for the sentence-level experiments², and one that contains the code for the site³ (i.e., a site that allows the user to insert a textual utterance and predict its emotion). Note that the repositories will only be public after the discussion of this M.Sc.

1.6 Master Thesis Structure


This thesis is organized as follows: Chapter 2 presents the fundamental concepts (Section 2.1) and related work (Section 2.2). Chapter 3 details the thesis proposal explaining the representation of words and text in a multilingual space, and the models proposed to infer emotion ratings for both words and textual utterances. Chapter 4 describes the experimental evaluation, presenting the datasets alongside the evaluation metrics, followed by the experimental results. In the end, Chapter 5 states the conclusions of this work and the possible future directions.

¹<https://github.com/SofiaAparicio/Sentiment-Analysis-words>

²<https://github.com/SofiaAparicio/Sentiment-Analysis-txt-utterances>

³<https://github.com/SofiaAparicio/Site-Thesis>

Fundamental Concepts and Related Work



This chapter is divided into two sections. The fundamental concepts are presented first, followed by related work, with a detailed description of the state-of-the-art on sentiment analysis and emotion quantification.

2.1 Fundamental Concepts

This section describes all the fundamental concepts necessary to understand the rest of the document. First, in Section 2.1.1, we have a description of different methods to encode text to serve as input to machine learning models. Section 2.1.2 presents a quick introduction to neural networks, presenting their different variants and their application in natural language processing. Finally, Section 2.1.3 explains the different methods that allow emotion representation.

2.1.1 Encoding Textual Information

The input given to a ML model for NLP cannot be a string directly. It is necessary to translate the text into numeric representation. Fortunately, there are several methods for doing this, and some of them will be described below.

First, consider the following sentence: *The dog and cat are on the car.* We could represent this sentence by a vector with dimensionality equal to the length of the sentence, i.e. eight numbers. Each word is then represented with a vector of seven 0s, except the position of the sentence where the word appears, that will be set to 1. This method is called a one-hot representation.

However, in the previous one-hot representation, it would be too expensive to represent a high dimensionality text. For example, in a text with 1000 words, each word we would have a vector of 999 zeros and an one, which would result in a computationally expensive and vulnerable to overfitting representation (Bengio et al., 2003). The correlation between words would not be captured, and specially the correlation of similar words (e.g. the words *dog* and *cat* do not have a notion of similarity in this representation). The necessity to provide an input that captures the

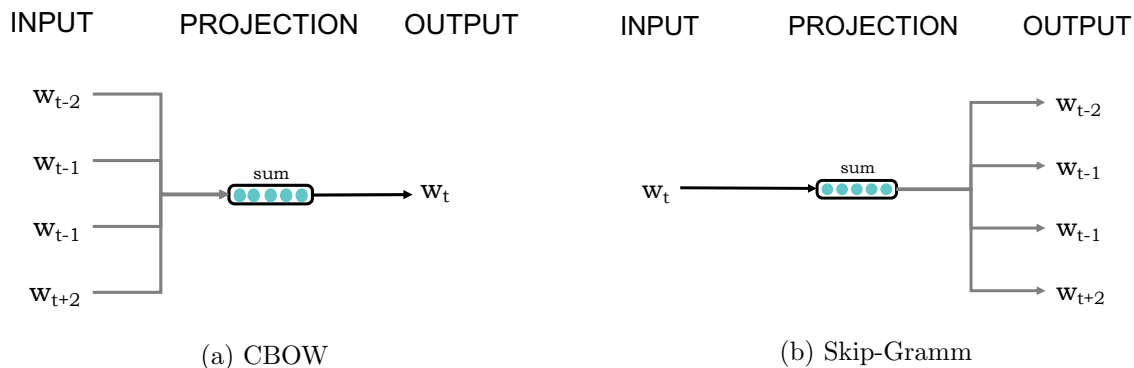


Figure 2.1: Dense word representations in a vector space.

similarity and intertwined relation between words leads us to the next approach, often referred to as a dense representation.

Dense vectors are composed by a smaller number of parameters, with values considering the surrounding environment of a word, and its connections with other words. The size of the vectors ranges, depending on the detail of the embedding. An embedding with a higher dimensionality can capture detailed connections between words. However, it is necessary to provide more training data so that procedures used to infer if the representation can work properly.

There are two common ML models for learning word embeddings. These are the Continuous Bag of Words (CBOW) and the Skip-Gram approaches, both illustrated in Figure 2.1.

A CBOW model predicts a central word w_t , based on the continuous distribution of the context (Mikolov et al., 2013a). It considers all the words in a distance d and combines all these surrounding words ($w_{t-d} \dots w_{t-1}, w_{t+1} \dots w_{t+d}$) to predict the central word. Equation 2.1 can translate how the model works, where T represents the number of words.

$$\frac{1}{T} \sum_{t=1}^T \log p(w_t | w_{t-d} \dots w_{t-1}, w_{t+1} \dots w_{t+d}) \quad (2.1)$$

The other approach, Skip-Gram, has the opposite behaviour of the CBOW. It predicts the context based on the centre word of the sentence (Mikolov et al., 2013b). Formally, it is defined by Equation 2.2, where d is the size of the training context.

$$\frac{1}{T} \sum_{t=1}^T \sum_{-d \leq j \leq d, j \neq 0} \log p(w_{t+j} | w_t) \quad (2.2)$$

The parameter $p(w_{t+j} | w_t)$ is the simple Skip-Gram probability, calculated by the softmax function. In its turn, v_{w_I} represents an input, and $v'_w{}^\top$ represents an output word, while W

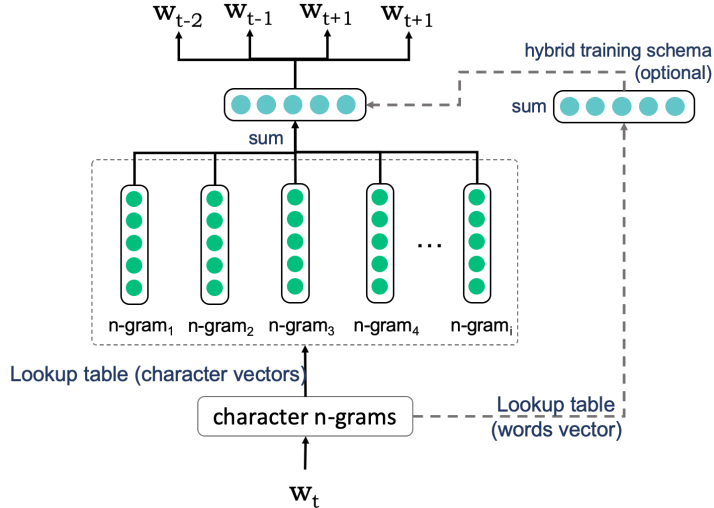


Figure 2.2: FastText

represents the number of words in the vocabulary.

$$p(w_O|w_I) = \frac{\exp(v'_{w_O}{}^\top v_{w_I})}{\sum_{w=1}^W \exp(v'_w{}^\top v_{w_I})} \quad (2.3)$$

Comparing both models (Mikolov et al., 2013a), we can consider Skip-Gram to work well with uncommon words and with less training data. Since CBOW is conditioned to the context, it is better to predict frequent words. For example, in the sentence *the girl is remarkable*, if we were predicting the word *remarkable* with the context *the girl is*, probably the system would predict *beautiful*, since it is a more common word. CBOW also needs more training examples to perform comparably to the other model. However, it is faster to train, since in Skip-Gram it is expensive to calculate the $\log p(w_{t+j}|w_t)$ for each word.

An approach that can use both CBOW and Skip-Gram inference, although more commonly used with Skip-Gram since it shows better results, is Word2vec. Word2vec (Mikolov et al., 2013a) receives a text and produces a vector space in which each word is assigned to vector in the space and similar words occupy close spatial positions. It should nonetheless be noted that, this model does not handle Out-Of-Vocabulary (OOV) words, i.e. words that were never seen in the training data.

A solution to the aforementioned problem appeared with FastText (Bojanowski et al., 2016; Joulin et al., 2016). This technique proposed the construction of word embeddings by adding morphological information to word2vec, thereby assigning distinct vectors for each part of a word. The authors proposed the use of a n-gram based model, where each word embedding is

the sum of all its n-grams. In their tests, the authors considered all n-grams where n was greater than 3 and smaller than 6. An example is given in Figure 2.2 for the word *hello*, that can be translated into Equation 2.4, where \mathbf{u}_w is the word vector of w and \mathbf{z}_g assumes the n-grams with as size g .

$$\mathbf{u}_w = \sum_{g \in \mathcal{G}_w} \mathbf{z}_g \quad (2.4)$$

2.1.2 Introduction to Neural Networks and Deep Learning

A Neural Network (NN) is a biologically inspired approach that is commonly used to address supervised classification problems, mimicking the biological brain. The main goal is to submit the input information to several operations, processing and structuring the data in a way that allows comprehension in a mathematical way, simulating what our brains do through synaptic connections between the neurons. These architectures were applied in the field of NLP to solve problems such as translation (Klein et al., 2017), sentiment analysis (Dos Santos and Gatti, 2014), question answering (McCann et al., 2017), among others.

The most rudimentary NN is called the Perceptron, featuring a single neuron. This linear model learns by taking a set of inputs, x_1, x_2, \dots, x_n , multiplying them by a set of weights, w_1, w_2, \dots, w_n , and adding a bias, b (McCulloch and Pitts, 1943), to produce an output y . The model can be translated into Equation 2.5 which also features an activation function, g .

$$y = \sum_{i=0}^n g(\mathbf{w}_i \times \mathbf{x}_i + b) \quad (2.5)$$

The role of the activation function is to determine how suitable a neuron is to a given output (i.e. determine if a neuron should be fired or not). A characteristic of activation functions is the necessity for a quick and efficient computation, since the model needs to adjust the parameters of the activation function several times for the model to learn, mainly because of the backpropagation learning algorithm (short for backwards propagation of errors).

Three different types of activation functions are commonly used in distinct problems. The most simplistic activation function, step-wise activation (Ng et al., 1997), is applied to provide binary results, not supporting an output with multiple values. The linear activation function provides multi-value outputs. The more complex activation functions, capable of more complex operations and generating better results (Specht, 1990), are nonlinear. The sigmoid function,

varying between 0 and 1, is useful to predict probabilities, although it can learn slowly if the values are too near of 0 or 1. The Rectified Linear Unit (ReLU), one of the most used activation functions, because of its fast computation, can be translated into Equation 2.6.

$$R(z) = \max(0, z) \tag{2.6}$$

Perceptrons have limitations, in the sense that they can only learn linear separable problems, and most data cannot be detachable linearly (Minsky and Papert, 1969). To solve more complex problems, it was necessary to concatenate several perceptrons into structures that are called a MLPs, or feedforward neural networks. Usually, this type of networks is composed of at least three types of layers: an input layer, one or more hidden layers, and an output layer. On a given MLP, the neurons are fully connected to each other to allow the flow of the information through the network, as we can comprehend through Figure 2.3 and Equation 2.7.

$$f(x) = g\left(b^{(2)} + W^{(2)}\left(s\left(b^{(1)} + W^{(1)}x\right)\right)\right) \tag{2.7}$$

To allow the network to learn formally, the weights W and bias b related to the synaptic connections require a constant adjustment, monitoring the loss function. The primary goal of this function is to determine how well the algorithm is modelling the training data. If the loss function returns an output number that is high, the model performed poorly, but on the other hand, if the model yields a low value, the model is operating well. The goal of training a model is to minimise the loss across different examples.

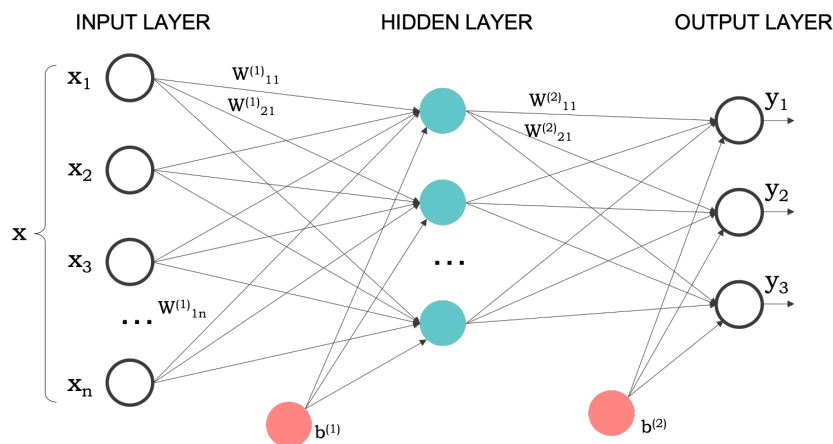


Figure 2.3: Multi-Layer Perceptron

The most used loss functions to train NLP models are generalized margin losses, which use a margin separating the correct answers and incorrect ones (LeCun et al., 2006), and the log loss (Roy and McCallum, 2001). Another approach is the negative log-likelihood loss, which uses probabilistic modelling. Applications of this method are the categorical cross-entropy loss (Covington et al., 2016) and several ranking losses (Goldberg, 2016).

When the loss function shows a poor performance, it is necessary to readjust the values of the weights through backpropagation. As the name suggests, backpropagation calculates the gradient of the loss function of a given input-output, using a variation of gradient descent (Ruder, 2016), and then the weights are updated to minimize the loss. When choosing one of the gradient descent variations, it is necessary to consider the amount of data, and the time we want to spend.

Vanilla or batch gradient descent computes the gradient of the cost function for the entire training dataset, as it is possible to observe through Equation 2.8. In order to perform one update of the weights all the weights, i.e. all the layers, it is necessary to calculate the gradient for the entire dataset, which is computationally expensive.

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta) \tag{2.8}$$

An opposite approach corresponds to Stochastic Gradient Descent (SGD), requiring the computation of the gradient for each training instance. In large datasets, the alterations of the weights can be redundant. This approach can be expressed through Equation 2.9.

$$\theta = \theta - \eta \cdot \nabla_{\theta} J\left(\theta; x^{(i)}; y^{(i)}\right) \tag{2.9}$$

A balance between both previous methods is mini-batch gradient descent, that computes the gradient considering n training examples. The procedure can be translated into Equation 2.10. It is much faster than Vanilla gradient descent, since it only calculates the gradient for a limited amount of data, and prevents redundant calculations, in larger datasets, for similar examples of training data. This method of calculating the gradient is the most used and, usually, the term SGD is also used to indicate the mini-batch gradient descent.

$$\theta = \theta - \eta \cdot \nabla_{\theta} J \left(\theta; x^{(i:i+n)}, y^{(i:i+n)} \right) \quad (2.10)$$

SGD still has one limitation. When converging, it can be faced with local minimum and decide it reached the best solution. Momentum is used to address this situation, speeding the process and decreasing oscillations. Among modern optimization algorithms we have the well known, Adagrad and Adam algorithms (Kingma and Ba, 2014).

MLPs allowed the progression of NNs. Yann LeCun, inspired by a model for the human visual cortex by Hubel and Wiesel (1962), developed the Convolution-and-Polling architecture (LeCun et al., 1995), also known as CNN. LeCun applied these techniques to images, and it was years later that CNNs were first applied to NLP. The first studies were conducted by Collobert et al. (2011) in the area of semantic-role labelling, and later studies by Kalchbrenner et al. (2014) and Kim (2014) focused in the fields of sentiment analysis and question-type classification.

The architecture of a CNN is generally composed of two parts, namely a set of a convolution + pooling of layers followed by several fully connected layers. In its turn, each convolution + pooling layer is divided into two operations, convolution and polling.

Firstly, in convolution procedures, we will have several types of filters, also called kernels. In each kernel, a determined characteristic of an input text is being searched for. The kernel size can vary with a number of words that we are interested in searching. A sliding window will determine which part of the sentence will be considered as a selective field, i.e. the field that will be analysed. When the kernel is passing through the selective field, it will calculate the dot product, producing an activated region that stores the detected characteristics in that region, as searched by a determined kernel. After having all the regions of the sentence activated by several kernels, we will have an activation map, that stores all the activated regions. These regions manifest specific features highlighted by the kernels.

Mathematically, the convolution layer can be translated by Equation 2.11. Consider a sequence of words $\mathbf{x} = x_1, \dots, x_n$, each word with the correspondent embedding vector $E\mathbf{e}(x_i)$. A convolutional layer with a width k , applying a moving window with the same size k over the sequence, will generate several instances of windows $\mathbf{w}_i = [x_i, \dots, x_{i+k-1}]$. Latter, a filter in the form of a regular linear function is applied to each window of the sequence. The filter is represented by a matrix \mathbf{F} , composed by a set of l different filters $\mathbf{f}_1, \dots, \mathbf{f}_l$. At the end, a bias b is added and an activation function g is applied element-wise.

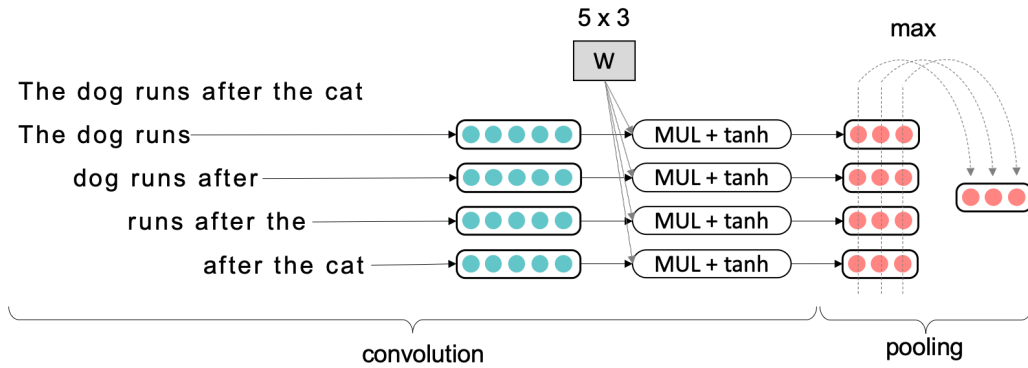


Figure 2.4: Convolution and pooling applied to the sentence *the dog runs after the cat*.

$$\mathbf{p}_i = g(\mathbf{w}_i \cdot \mathbf{F} + \mathbf{b}) \quad (2.11)$$

Afterwards, it is necessary to condense all the information extracted by the kernels, since it would be computationally exhausting to save all the information previously calculated. The pooling operation is responsible for reducing the spatial dimension of the activation map. This operation will not affect the depth dimension of the volume, because it extracts the most salient information. There are several studied ways of performing this type of downsampling, but the most used is max pooling. In Equation 2.12 we denote that the vectors $\mathbf{p}_1, \dots, \mathbf{p}_i$ will be combined into a single vector $c_{[j]}$ that represents the entire sequence. All the transformations can be summarized in Figure 2.4, inspired on an original figure by Goldberg (2016).

$$c_{[j]} = \max_{1 < i \leq m} p_i[j] \quad \forall j \in [1, \ell] \quad (2.12)$$

When dealing with NLP, it is common to consider word sequences to represent a textual utterance. MLPs can accommodate sequences through vector concatenation and vector addition (CBOW), but the order of the information is discarded (Bebis and Georgiopoulos, 1994). When considering CNNs, they maintain some sensitivity to word order, corresponding to local patterns. However, these models discard the order of the whole sentence (Kalchbrenner et al., 2014).

When the order of the sentence is relevant, it is necessary to implement another strategy. Socher et al. (2013) describe one of the early applications of what we now call Recurrent Neural Network (RNN). These differ from the previous models by handling variable-length inputs.

The most straightforward application of this idea is called the Simple Recurrent Neural Network (S-RNN). The main difference from techniques that apply CBOW is appending a

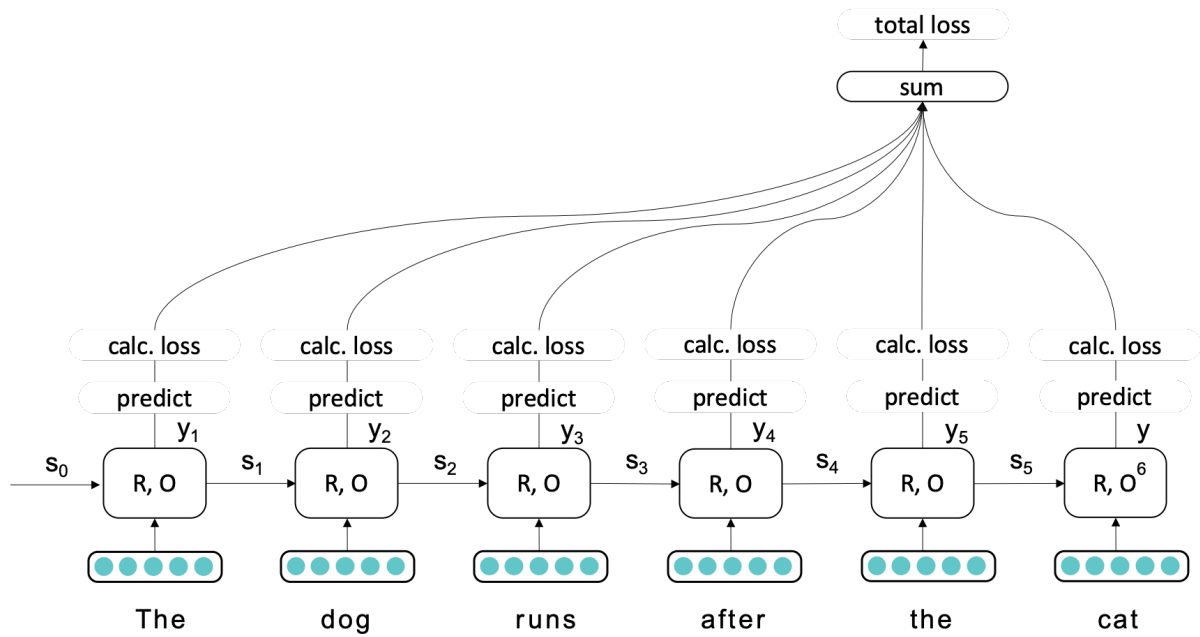


Figure 2.5: Summarizing the behaviour of a RNN.

linear transformation followed by a nonlinear transformation. This provides the network with the knowledge of previous outputs. Thus, the new output is moulded by previous information, providing memory to the network (Elman, 1990).

Mathematically, when training an RNN, each hidden state x_t at a given time t is the output given by an input sequence x_t at the time t and all the previous input states s_{t-1} adding a bias b . Afterwards, an activation function is applied. This can be translated into Equation 2.13.

$$s_t = g(x_t \cdot w_1 + s_{t-1} \cdot w_2 + b) \quad (2.13)$$

In this structure, we have the functions R , that allows keeping track of the state through the vector state $s_t = R(s_{t-1}, x_t)$, and O , defined by $y_t = O(s_t)$ that provides the output over which we can calculate a loss. Each different RNN structure will require different instantiations of these functions. The Simple Recurrent Neural Network approach can be summarised in Figure 2.5.

Training the aforementioned model can be associated to problems, and a common obstacle is the increase of the norm of the gradient, also called vanishing gradient.

Vanishing gradients is a consequence of the incapability of later steps to reach initial inputs, due to quickly diminishing values. In each step, previous output are considered. The calculated gradient is not only referring to the current node but also all the previous ones. When making improvements, by backpropagation, the corrections to previous layers will be smaller each time,

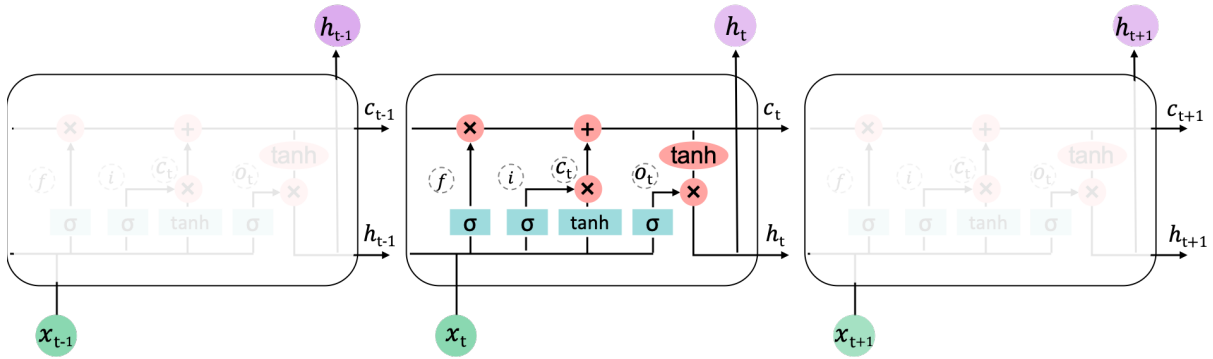


Figure 2.6: Summarizing the behaviour of a LSTM.

being impracticable to have long-term interdependencies (Pascanu et al., 2013).

Owing to the vanishing gradients problem, that results in the incapability of S-RNNs to learn long-range temporal dependencies, it was necessary to create models that could provide long-term memory: LSTM and Gated Recurrent Unit (GRU). These two models solve the vanishing gradient problem by providing more controlled memory access through a gated architecture. Gated architectures determine which part of the memory should be disregarded and which part of the new input will be stored in the available space. Gates can be composed by sigmoid activations, that limit the values between 0 and 1, and the hyperbolic tangent function.

Hochreiter and Schmidhuber (1997) proposed the first modification of simple RNNs to solve the vanishing gradient problem: the LSTM. RNN architectures use a recursive function R , that can also be called state vector, and that that function encodes a determined $x_1 : n$ sequence.

LSTMs also use a state vector but splits it in two: the part responsible for the working memory and the memory cell, where the essential parts of the sequence are stored. As was previously stated, this architecture differs from simple RNNs because of its gated architecture. In the case of LSTMs, we have to consider three types of gates.

First, we have a forget gate f , responsible for determining what information should be kept. Second, an input gate i combines the previous hidden state and the current input and selects values that should be updated, through a sigmoid function. Cell gates c_t are the next stage, doing a pointwise addition that returns a new state cell with the new values that the network will compute. Ultimately, the output gate o decides what should be carried to the next hidden state. This step combines the new state and the memory cell. The LSTM architecture can be summarized in Figure 2.6 and Equation 2.14.

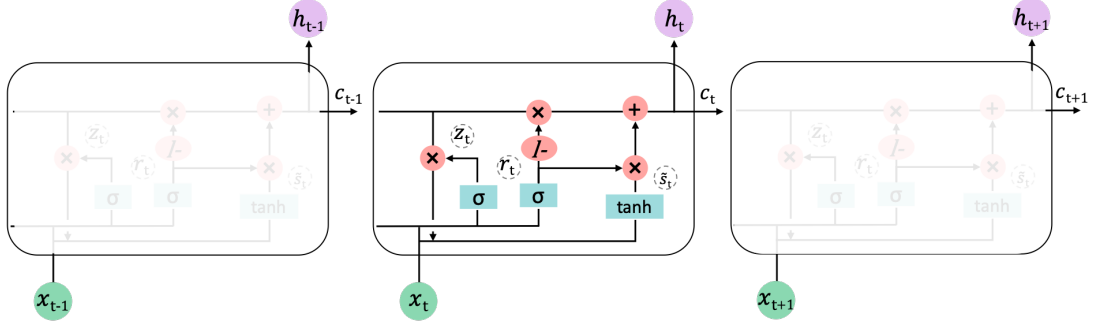


Figure 2.7: Summarizing the behaviour of a GRU.

$$\begin{aligned}
s_t &= R_{\text{LSTM}}(s_{t-1}, x_t) = [c_t; h_t] \\
c_t &= f \odot c_{t-1} + i \odot z \\
h_t &= o \odot \tanh(c_t) \\
i &= \sigma(x_t W^{xi} + h_{t-1} W^{hi}) \\
f &= \sigma(x_t W^{xf} + h_{t-1} W^{hf}) \\
o &= \sigma(x_t W^{xo} + h_{t-1} W^{ho}) \quad g = \tanh(x_t W^{xz} + h_{t-1} W^{hz}) \\
y_t &= O_{\text{LSTM}}(s_t) = h_t
\end{aligned} \tag{2.14}$$

Cho et al. (2014) proposed a new alternative to LSTMs, namely the GRU. It performs comparably to LSTMs but requires fewer gates, and it eliminates the need for a separate memory. It only requires two gates namely an, update gate and a reset gate. The update gate operates similarly to the input gate on the LSTM. It selects the information that will be discarded and what will be apprehended. The reset Gate r manages the information that should be forgotten, as it can be observed through Equation 2.15 and Figure 2.7.

$$\begin{aligned}
s_j &= R_{\text{GRU}}(s_{j-1}, x_j) = (1 - z) \odot s_{j-1} + z \odot \tilde{s}_j \\
z &= \sigma(x_j W^{xz} + s_{j-1} W^{sz}) \\
r &= \sigma(x_j W^{xr} + s_{j-1} W^{sr}) \\
\tilde{s}_j &= \tanh(x_j W^{xs} + (r \odot s_{j-1}) W^{sg}) \\
y_j &= O_{\text{GRU}}(s_j) = s_j
\end{aligned} \tag{2.15}$$

Despite the aforementioned improvements, RNNs and their variants still have disadvantages

(i.e. difficulty in capturing dependencies between the words of a sentence if they are too distant from each other (Shen et al., 2018)). For that this reason, attention mechanisms were proposed to address parts of these problems.

Attention was introduced by Bahdanau et al. (2015) to solve translation. They proposed the use of a layer that gives attention to each source sentence word and determines which words are more relevant to achieve the expected output, even when the sentences are reasonably long. In other words, the decoder receives an additional weighted input that determines which tokens are necessary to pay more attention, in each time step.

This model was originally composed by an encoder-decoder mechanism. The encoder is responsible for processing the input sequence, treating it through encoding mechanisms (i.e. summarizing and shortening the sequence) until it is transformed into a single context vector with a fixed size. Then, this representation is passed to the decoder, where the vector is transformed to the desired output. The vector passes through a feed forward NN, using a softmax function to compute the attention weights. A context vector is then computed, and then this vector is concatenated with the context vector of the previous time step. In the end, the output word is delivered.

The attention mechanism got its name from the capacity of looking at a textual sequence and generate relationships between words, even though they might be distant from each other. It gives attention to the relevant words and ignores the words that are not very relevant. There are several types of attention related to different categories, discussed by Chaudhari et al. (2019). In the thesis, the main focus will be on multi-head self-attention.

Vaswani et al. (2017) showed that self-attention mechanisms are not only companions of other well-known machine learning models, but they can also be used independently of other mechanisms. The authors proposed the Transformer, a learning-based translation mechanism based on multi-head self-attention. The model outperformed previous approaches, while also having a faster training time.

2.1.3 Emotion Representation

In sentiment analysis it is crucial to assign a determined feeling towards a word or a larger textual utterance. Two approaches originally proposed in the cognitive sciences can be used to define how emotions can be delimited and perceived by humans (Ekman and Friesen, 1971).

One of these approaches defends the existence of six basic universal emotions: happiness,

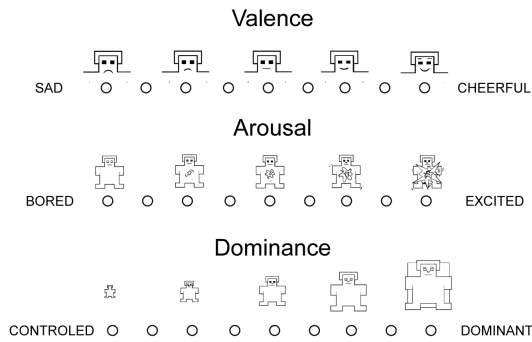


Figure 2.8: Self-Assessment Manikin

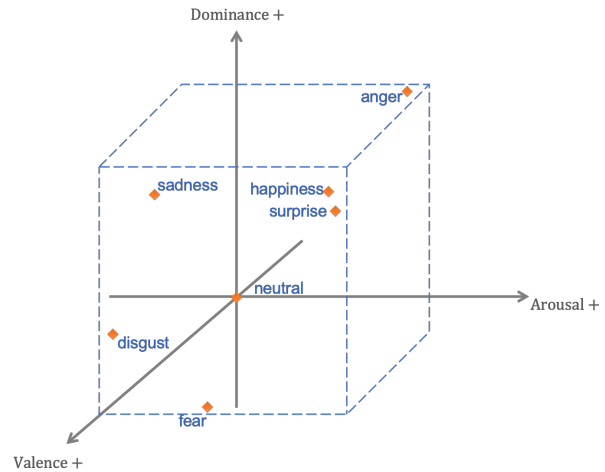


Figure 2.9: Valence, arousal and dominance three-dimensional space

anger, sadness, surprise, disgust, and fear (Ekman, 1992). This is called the categorical approach. The other approach, designated by dimensional, advocates continuous numerical values that progress through multiple dimensions (Russell, 1980). After the original dimensional approach, other models were suggested (Wang et al., 2016). Studies were conducted with the goal to collect properties of words, like complexity, frequency, or neighbourhoods, among others (Kuperman et al., 2012).

The emotion representation that was used in this thesis is inspired by an early study, conducted at the University of California by Bradley and Lang (1999). The study produced a normative emotion rating of valence, arousal, and dominance for 1034 English words. It assessed these three dimensions by the Self-Assessment Manikin (SAM) questionnaire (Lang, 1980) and the manikin is a visual representation of three dimensions (Russell and Mehrabian, 1977). The SAM and the meaning of the three dimensions can be represented through Figure 2.8. The three dimensions were rated on a scale from 1 to 9, as shown in Figure 2.9. Warriner et al. (2013) extended the ANEW study by presenting a lexicon with 13 915 English words.

The dimensional representation is considered to better model emotions than the categorical approach. As we can observe in Figure 2.9, the six basic emotions are not even considered to be opposite references on the dimensional representation.

2.2 Related Work

In recent years, the area of computational linguistics became interested in some of the emotion norms collected in behavioural psychology, because of their usefulness for categorizing

texts according to the sentiments and emotions being expressed. However, to evaluate the emotional ratings (i.e. valence, arousal, and dominance) of textual utterances it requires an available large quantity of rated words. Alongside the limited amount of datasets available and by the size of these datasets of affective norms (Tang et al., 2014). This is particularly true if we consider languages other than English, or applications that rely on the analysis of short sentences (e.g. Twitter messages). This lead researcher to develop new studies for collecting human emotional ratings for large sets of words, e.g. through crowdsourcing methodologies (Bradley and Lang, 1999; Årup Nielsen, 2011; Scott et al., 2019; Warriner et al., 2013), or to examine whether affective norms can be calculated using automatic procedures, such as machine learning.

This section has an overview of studies that I found relevant to mention considering the goals of this thesis. It is organized as follows: Section 2.2.1 describes studies where emotion ratings are assigned to words, and in Section 2.2.2 studies assigning emotion ratings to textual utterances.

2.2.1 Assigning Sentiment to Words

Assembling human ratings for words is expensive and difficult to make, particularly if we consider different languages. Given a small seed lexicon with emotional rating, a number of techniques have been proposed to automatically estimate ratings for new words. For instance, regression-based methods to automatically calculate unrated words from previously ranked words was developed, primarily considering the proximity (length, contextual diversity, co-occurrences) between a word previously rated and a non-rated word (Recchia and Louwerse, 2015; Köper and Im Walde, 2016).

Bestgen and Vincze (2012) used Latent Semantic Analysis (LSA), a process to evaluate what words are more relevant to the textual corpus to assign better the values of the valence, arousal, and dominance of the words. In this context, LSA receives an input matrix (i, j) , with the size of the words i by the documents j . In each entry, it is determined the number of times each word appears in each document. In order to decrease the influence of the most frequent words, each term is weighted, and Singular Value Decomposition (SVD) is applied to factor the matrix into three new matrices \mathbf{U} , \mathbf{S} and \mathbf{V} . The product of these matrices yields the original matrix. A new matrix, with a low-dimensional similarity to the original matrix, can be retrieved by trimming down the original matrix to a fixed number of dimensions, before calculating the product. Lastly, to identify the similarity between two words, it is necessary to compute the cosine between their corresponding rows. It is necessary to note that, as long as two words

appear in similar documents, they may show a high cosine, because of the rank reduction. LSA can be considered to be a measure of higher-order co-occurrence since it links words that appear in similar linguistic contexts.

Bestgen and Vincze used the previous method to compute the association of each of the 17,350 words in the Touchstone Applied Science Associates (TASA) dataset. This dataset is constituted by ten million tokens (92,409 types) of high-school level English text. They also estimated the valence, arousal, and dominance of each word by computing the mean value of each dimension using their thirty closest neighbours (excluding each word itself, i.e. using leave-one-out cross-validation and relying on a k nearest neighbour interpolation technique). The obtained estimates achieved a Pearson’s correlation of 0.71, 0.56 and 0.60 with the ANEW norms on valence, arousal and dominance, respectively, using a set of 953 words that were present in both the ANEW norms and in the TASA corpus. When analyzing the results, the authors also reported that one of the problems with the proposed method relates to the fact that word co-occurrence models, including LSA, often model antonyms as close neighbours in a vector space, thus sometimes calculating the wrong predictions (Bestgen and Vincze, 2012).

Other methods were also tested, like Hyperspace Analogue to Language (HAL) (Lund and Burgess, 1996) that considers only a surrounding window of words preceding and following the targeted word, usually no more than ten surrounding words. Moreover, a skip-gram model with negative sampling was later introduced (Mikolov et al., 2013a). Alike LSA performance, except for scalability since skip-gram only increases linearly. This model is also mentioned as word2vec. Mander et al. (2015) compared both HAL and skip-gram models with different extrapolation techniques, although they concluded that these methods lead to different results than ratings made by humans.

Recchia and Louwse (2015) noted that several previous studies (Citron et al., 2014; Jamin and Casasanto, 2012) have shown correlations between emotional ratings and other lexical variables (e.g., word frequency, word length, or orthographic similarity). Recchia and Louwse thus attempted to further improve results, by integrating into the prediction models additional variables that in the literature have been shown to correlate with valence, arousal, or dominance, as well as additional variables that can contribute to an independent variance.

First, the authors attempted to replicate the study by Bestgen and Vincze, using a more scalable approach based on Point-wise Mutual Information (PMI). Leveraging a larger Web corpus to estimate the word co-occurrence statistics, and using all 12,764 words in the set of norms from Warriner et al. (2013) that did not occur in the ANEW corpus, as training data for

predictive models. Specifically, Recchia and Louwerse used bigrams and trigrams appearing in the Google Web 1T 5-gram corpus to compute co-occurrences, within a window size of two, between each word in the Warriner dataset and all the words in the training dataset. The co-occurrences were then used to compute word similarities based on positive PMI cosines, according to the process specified by Bullinaria and Levy (2007) (i.e. each word is represented by a vector of n (n corresponds to the number of words present on the vocabulary) elements, and each element of each vector corresponds to the PMI score between the target word and the word that is indexed by the particular element of the vector. The elements that contain negative values are set to zero, and the cosine between the vectors is used to compute word similarity). For the words present on the training set, the authors determined its k nearest neighbours using the positive PMI cosine measure, and they then calculated the mean valence, arousal, and dominance of these k words. The authors measured a Pearson's correlation of 0.74, 0.57 and 0.62 to ANEW valence, arousal and dominance ratings, respectively. These values were achieved by using values of 15, 40 and 60 for the k parameter, respectively (Recchia and Louwerse, 2015).

In the second set of experiments, Recchia and Louwerse attempted to see whether including additional variables in a linear regression model would improve the predictions for the affective ratings of words (Recchia and Louwerse, 2015). Specifically, the authors considered variables such as the log frequency of the word, its contextual diversity, the word length, and the mean valence, arousal and dominance of the word's nearest semantic neighbours, according to positive PMI cosines or according to several measures of orthographic similarity. Besides the aforementioned features, the authors also considered *right-side advantage* scores for the words as proposed by Jasmin and Casasanto (2012), by subtracting the number of letters in the word that appear in the left-hand side of the keyboard, from the number of letters that appear in the right-hand side. Three linear regression models were fit in a greedy step-wise fashion, respectively with valence, arousal, and dominance as the dependent variables. The authors measured Pearson correlation coefficients of 0.80, 0.62 and 0.66, respectively for the ANEW norms of valence, arousal, and dominance. By considering additional variables, the authors have thus managed to significantly improve the results, up to the level where the correlations closely resemble those that are obtained from different human judges.

Recchia and Louwerse also reported on an initial analysis that investigated whether predictions can be made for languages other than English, leveraging the Spanish and Dutch versions of ANEW as the source norms of valence, arousal, and dominance, together with the Dutch and Spanish versions of Wikipedia for the computation of word co-occurrences (Recchia and

Louwerse, 2015). At 4,299 and 1,034 words each, these datasets provide only a fraction of the English data provided by Warriner et al. (2013), significantly reducing the amount of information available to learn predictive models, for each target language separately. Nonetheless, through a leave-one-out cross-validation methodology, the authors report on encouraging results for the method based on the nearest semantic neighbours (i.e., a Pearson’s correlation of 0.52, 0.36 and 0.48 in Spanish, and of 0.50, 0.47 and 0.37 in Dutch, respectively for valence, arousal, and dominance).

Mandera et al. (2015) also researched the usage of textual dataset to build a semantic similarity space, latter applying ML techniques to extrapolate existent ratings to unrated terms. The authors conducted a systematic comparison of two extrapolation techniques (i.e. kNN and random forest regression), in combination with semantic spaces built from an English subtitle corpus including approximately 385 million words, and leveraging different vector representations for the words. These include representations build through (i) LSA, (ii) a generative topic model known in the literature as Latent Dirichlet Allocation (Blei et al., 2003), (iii) a representation based on PMI similar to that from the study by Recchia and Louwerse (2015), and an approach leveraging neural networks that is commonly referred to as word2vec’s skip-ngram model, similar to the one that is used here. A method based on the k nearest neighbours, leveraging the skip-ngram word embeddings, resulted in the most accurate predictions, although significantly inferior to those from previous studies Recchia and Louwerse (2015); Bestgen and Vincze (2012) (i.e., correlations of 0.694, 0.478 and 0.595 in 10-fold cross-validation experiments with the ratings from (Warriner et al., 2013), respectively in terms of valence, arousal and dominance). The authors nonetheless argue that the random forest method has the advantage of more easily being able to incorporate additional predictors. The authors also state that the higher correlations obtained using the skip-ngram model, in comparison to their other approaches, can perhaps be explained by the fact that this method is better at estimating word similarities (Baroni et al., 2014).

Sedoc et al. (2017) developed an approach to distinguish words that are on opposite sides of the rating scale but share similar vector representations. First, the distributional hypothesis is leveraged, and words appearing in similar contexts have similar scores. Nonetheless, words that appear in similar contexts but have opposite polarities or ratings will still have similar scores. Therefore, a second step is added. To detect similar un-rated words, signed spectral clustering (SSC) (Sedoc et al., 2016) is used. SSC combines regular spectral clustering (Ng et al., 2002) with additional information by negative edges, thus repealing words with different scores from the same clusters. Formally, the method can be translated to Equation 2.16, were $\text{vol}(A_j)$

represents the similarity between all nodes of the graph, $\text{links}^-(A_j, A_j)$ represents negative edges within the cluster, and $\text{cut}(A_j, \overline{A_j})$ are normalized clusters.

$$\sum_{j=1}^k \frac{\text{cut}(A_j, \overline{A_j}) + 2 \text{links}^-(A_j, A_j)}{\text{vol}(A_j)} \quad (2.16)$$

The methods were tested in three different languages to obtain affective norms of valence and arousal. Experiments were made with the English lexicon with 13,915 words from Warriner et al. (2013), the Spanish lexicon with 14,031 words from Stadthagen-Gonzalez et al. (2017), and the Dutch lexicon with 4,300 of Moors et al. (2013). The results presented by the kNN (for English 0.684 and 0.551, for Spanish 0.657 and 0.447, for Dutch 0.557 and 0.544, respectively for valence and arousal) and regression methods (for English 0.751 and 0.547, for Spanish 0.677 and 0.203, for Dutch 0.566 and 0.545 all for valence and arousal, respectively), with a 10-fold cross-validation setup, correspond to good results. However, the results were outperformed by a multi-task learning neural network (MTLNN) technique (Buechel and Hahn, 2018). They produced an alteration to an MLP with 5 hidden layers, shared across VAD, and 3 units of output, each representing a VAD dimension. The model was tested on 9 typologically diverse languages using different types of embedding models.

In another study (Li et al., 2017), several affective meanings were extracted in a multidimensional model. Word embeddings were inferred through unsupervised learning, and the authors later provide small sets of words to a train ridge regressor and support vector regressor models. The authors concluded that each embedding carries not only semantic meaning of a word but also sentiment. In their study, the Bayesian ridge regression was the model that performed better. The tests were conducted with datasets considering several languages and dimensions, among which they tested the ANEW (Bradley and Lang, 1999) (correlation of 0.821 for valence, 0.979 for arousal and 0.988 for dominance), Warriner et al. (2013) (with correlation of 0.934 for valence, 0.991 for arousal and 0.989 for dominance) datasets in English and a Chinese dataset (Yu et al., 2016) (with 0.582 for valence and 0.803 for arousal, considering the dataset did not feature the dominance ratings).

Some previous studies also focused on the possibility of cross-language approaches for automatically generating lexical resources (Banea, 2013; Banea et al., 2013), for instance exploring methods for generating lexical resources for subjectivity analysis in a new language (e.g., Spanish or Romanian) by leveraging English tools and resources. Given a bridge between English

Experiment	Datasets Tested	Models	Results			
			Valence	Arousal	Dominance	Metric
Bestgen and Vincze (2012)	ANEW	LSA	0.71	0.56	0.60	Pearson correlation
	Warriner	PMI	0.74	0.57	0.62	
Recchia and Louwerse (2015)	Dutch 4 299 words Spanish 1 034 words	3 linear regressions	0.80	0.62	0.66	Pearson correlation
		co-occurrences with other languages	0.52	0.36	0.48	
			0.50	0.47	0.37	
Mandera et al. (2015)	Warriner	kNN	0.694	0.478	0.595	Pearson correlation
	Warriner		0.684	0.551	-	
Sedoc et al. (2017)	Spanish 14 031	kNN	0.657	0.447	-	Pearson correlation
	Dutch 4 299		0.557	0.544	-	
Li et al. (2017)	ANEW	Baysan Ridge Regression	0.821	0.979	0.988	
	Warriner		0.934	0.991	-	
	CVAW (Chinese)		0.582	0.800	-	

Table 2.1: Most important studies related to assigning sentiment to words.

and the selected target language, the proposed methods can be used to automatically generate resources for the new language. In one particular experiment, Banea et al. (2013) started by selecting a small set of seed words that were known to be subjective, from an English source language lexicon. These seed words were then translated into the target language. Afterwards, the authors expanded the lexicon formed by the translated seed words by solely using material in the target language, specifically through a bootstrapping mechanism that uses LSA in order to measure word similarity. The authors found that starting from a small number of manually translated seeds, in a target language, can instantly grow a subjective lexicon. And it proves to outperform a fully automatic translation of a fully developed lexicon in a source language.

All the previously stated studies are summarised in table 2.1.

2.2.2 Assigning Emotion to Textual Utterances

In emotion analysis, word-level prediction differs a lot from assigning emotion values to larger linguistic units, such as paragraphs and sentences. Binali et al. (2010) recognised three different approaches for emotion detection: keyword-based, learning-based, and hybrid. However, all these methods resort to different linguistic analysis tools (e.g., semantic level, sentence segmentation, parts of speech recognition, token level).

The first approach relies heavily on text preprocessing and relies on a domain specific theory, regarding several independent domains that hold different emotions. Thus, textual utterances are divided into words for the extraction of sentiment. Ma et al. (2005) uses keyword spotting applied to a chat system to generate emotionally responsive messages. Malheiro et al. (2016) make use of a keyword approach to analyse song verses, considering the valence and arousal space.

However, word-level problem solving cannot fully address high-level linguistic prediction

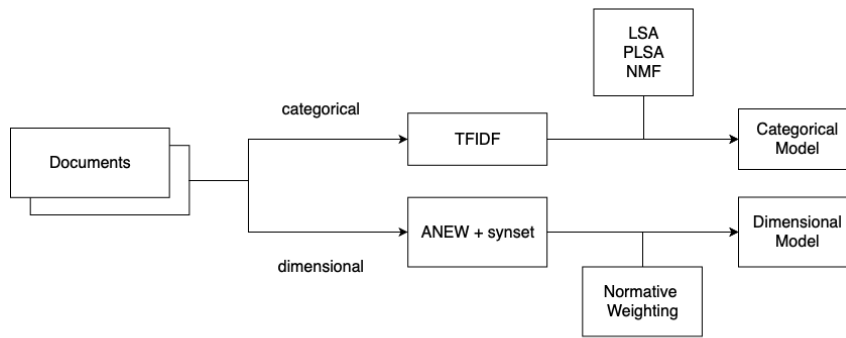


Figure 2.10: Method proposed by Calvo and Mac Kim (2013).

because of the way these words are combined (LaBrie and Louis, 2003). One example is handling negation or irony, which can change the meaning of the text, and ignored if considering the words separately. The second, learning-based, considers a set of training data to shape a predictive model. This approach falls into two different categories depending on how the input is organised (Buechel and Hahn, 2018). One is arranged spatially, such as architectures that use convolutional neural networks (CNN). The other uses sequential input data, typical for RNN, LSTM and GRU models.

Firstly, considering the input arranged spatially, we have an early study conducted by Calvo and Mac Kim (2013). In this study, the primary goals were to evaluate the two models of sentiment representation, namely the dimensional and the categorical models, and determine what could be their applications and what could be the expected accuracy. For the categorical model, the text was converted into a VSM representation with TF-IDF weights. The VSM representation can then be reduced with LSA, probabilistic LSA (PLSA) and, Non-negative Matrix Factorization (NMF). These three translate the pseudo-documents into predefined categories. In the dimensional model, the authors resorted to ANEW and WordNet synsets. Each word is converted to the ANEW affective space. Afterwards, the words can be used to weight the sentence emotional place, naively. These methods are summarized in Figure 2.10. The NMF approach and dimensional model outperformed the other two.

A few years later, Buechel and Hahn (2016) wanted to foretell the emotion of a linguistic unit by a fine-grained analysis, using a regression model instead of classification and using two metrics to validate their results (Pearson correlation and root-mean-square error). The authors mapped the two emotion representations, translating the VAD output into a BE representation. Even though this method reduces performance, it still outperforms former systems that consider the three dimensions.

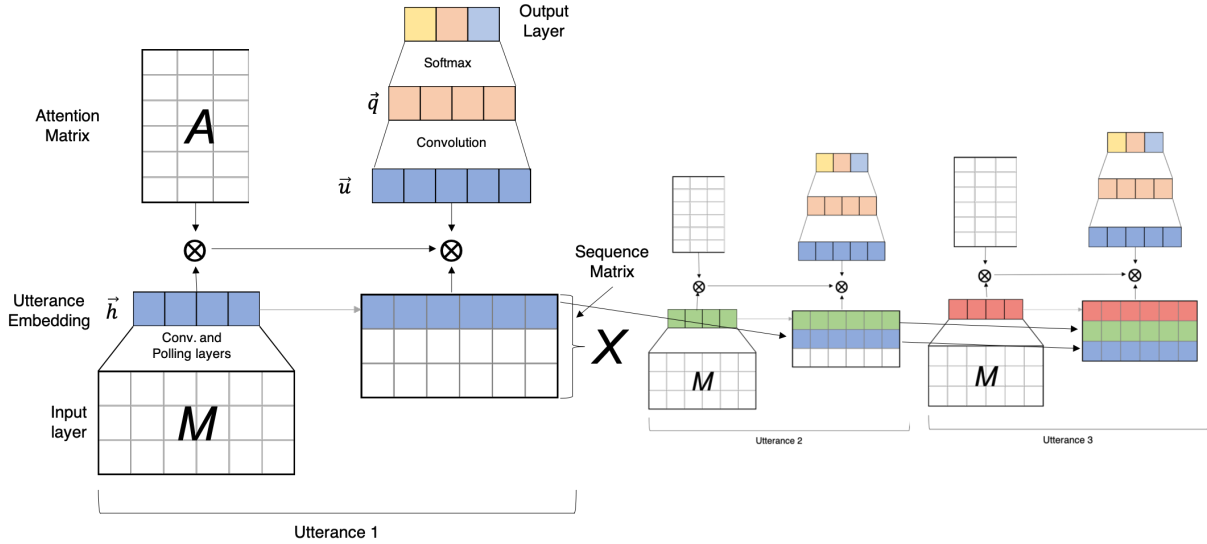


Figure 2.11: Sequence convolution neural networks unification by concatenation.

Since the amount of text documents rated in VA space is scarce, Preotiuc-Pietro et al. (2016) chose to resort to two psychologically-trained annotators. Facebook posts were rated, firstly, considering the valence and arousal dimensions separately. Afterwards, the experts asked to rate the two aspects together. In sum, 2895 messages were evaluated and VA parameters were compared through age and gender of the writer, with the authors concluding that female post writers express both more arousal and valence. Later, a two linear regression model using a BoW representation, on 10-fold cross-validation with this data, reaches a high correlation to the annotated results, obtaining a Pearson correlation of 0.650 and 0.850 for valence and arousal, respectively.

With the limited research on the use of sequential input data and the need for more emotionally rated data, Zahiri and Choi (2017) started a new investigation. The dialogues from the show TV Friends were annotated considering seven emotions: sad, mad, scared, powerful, peaceful, joyful, and neutral. Since CNNs are not ideal for processing sequences and RNNs perform slowly, the authors induced four sequence-based convolution neural networks (SCNN). The input for all SCNN is the same: a matrix M , with dimensionality equal to the number of tokens in any utterance by the embedding size. Each row in M represents a token in the utterance.

Comparing all the four models that were created, the model that performed better was the one represented in Figure 2.11. A matrix X is created by fitting attention matrix A to the current feature vector. The weights of A are adjusted considering past feature vectors.

Köper et al. (2017) introduced a test on a Twitter corpus from 2016 retrieved with emotion

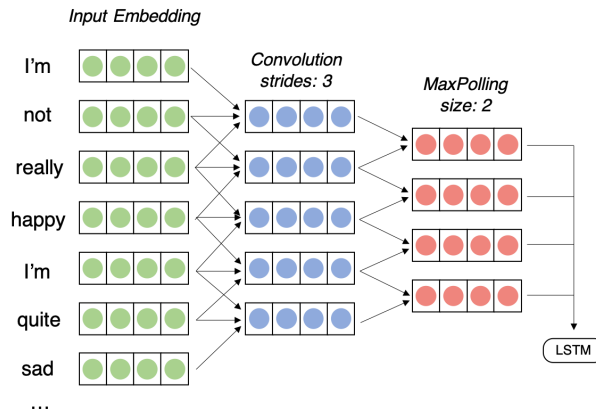


Figure 2.12: CNN and LSTM model proposed by Köper et al. (2017)

hashtags plus popular general hashtags. The authors selected only words with more than ten occurrences. Then, a combination of CNN and LSTM layers, represented in Figure 2.12, was used and trained. Later, the features were combined in a random forest classifier to estimate the result for each of the four emotions: anger, fear, joy, and sadness. To verify their results, the authors experimented different architectures: linear regression, MLP, two stacked LSTM. Analysing the results, the authors concluded that CNN-LSTM architecture outperformed the other three.

To remedy the struggle of learning emotion from a text through learning techniques, Kratzwald et al. (2018) proposed several modifications from previous models. Both categorical and dimensional emotion models were considered in an approach that combines a bidirectional LSTM (BiLSTM) layers, dropout layers for regularisation, and weighted loss functions to cope with the imbalanced distribution of labels. All of this is done by passing the documents through an embedding layer, transforming the one-hot encoding of words in a numerical representation according to its semantic meaning. All the experiments can be summarized through the diagram in Figure 2.13. The performance of the BiLSTM compared to the simple LSTM was measured in terms of the Mean Squared Error (MSE), using a dataset of Facebook post (Preoțiuc-Pietro et al., 2016). The results showed an error of 1.007 for valence and 3.519 for arousal, with the LSTM, and 0.990 and 3.550 for the BiLSTM. This BiLSTM outperforms more from traditional machine learning models, up to 23.2% in F1-score.

Akhtar et al. (2019) proposed a multi-task ensemble model. The main idea is to group the intermediate layer from three pre-trained models, a CNN, a LSTM, and a GRU, with and a feature representation layer trained to capture the connections between all the previous layers. This approach can perhaps reach better results by considering the hypothesis of four individual systems. After ensembling the models, an MLP with 4 hidden layers will provide the predictions.

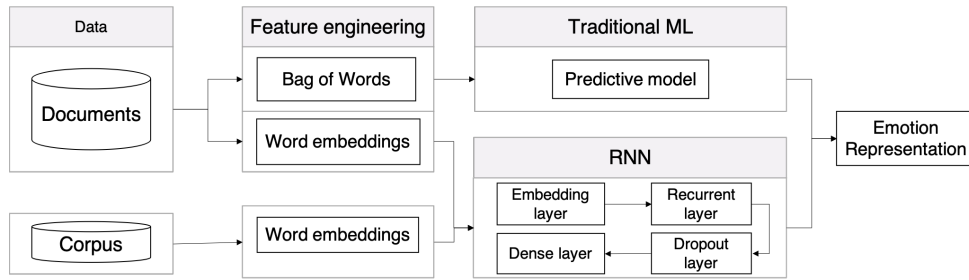


Figure 2.13: Pipeline for inferring emotion based on studies conducted by Kratzwald et al. (2018).

Following the trend of analysing emotion from social media, the authors also tested their model in datasets composed of Facebook posts (Preoțiu-Pietro et al., 2016) and Twitter posts (Buechel and Hahn, 2017). The authors obtained a Pearson correlation of 0.727 for valence and 0.355 for arousal for the Facebooks dataset, and 0.635, 0.375 and 0.277 for the EmoBank dataset (for valence, arousal and dominance respectively), with 10-fold cross-validation.

In a master thesis (Godinho, 2018) also applied a bi-directional RNN (BiRNN), followed by a max-pooling operation, to infer sentiment. The results, with the model trained with the Facebook posts dataset, showed good results when tested with ANET (Bradley and Lang, 2007) (Pearson correlation of 0.706 and 0.299, MAE of 1.963 and 4.575, MSE of 4.777 and 25.008 for valence and arousal, respectively). In the thesis, another model was also tested. An BiLSTM followed by max polling and an attention layer provided very promising results. It was tested with 10-fold cross-validation for EmoBank dataset (Pearson correlation of 0.553 and 0.348, MAE of 0.268 and 0.251, MSE of 0.127 and 0.104 for valence and arousal, respectively) and the Facebook posts (Pearson correlation of 0.725 and 0.925, MAE of 0.659 and 0.613, MSE of 0.743 and 0.650 for valence and arousal, respectively).

Buechel et al. (2018) used a 10-fold cross-validation technique to compare seven individual models (i.e. Ridge Regressor with two variations, Feed-Forward Network, GRU, LSTM, CNN, and CNN with an LSTM), considering datasets in several languages. The experiments were conducted for each dataset separately. Overall, GRU showed better performance, with a Pearson’s correlation of 0.74 for the Bradley and Lang (2007)(ANET) dataset, 0.57 for Imbir (2016b)(ANPST), 0.69 for Pinheiro et al. (2017)(MAS). These values were obtained conducting a mean of Pearson’s correlation of the three dimensions. Considering other models, the authors obtained similar results (i.e. for the CNN an 0.70 for ANET, 0.45 for ANPST and 0.62 for MAS; for the LSTM model an 0.73 for ANET, 0.56 for ANPST and 0.65 for MAS).

All the previously studies are summarized in Table 2.2.

Experiment	Datasets Tested	Models	Results			
			Valence	Arousal	Dominance	Metric
Preoțiuc-Pietro et al. (2016)	Facebook 2895 posts	BoW	0.650	0.850	-	Pearson correlation
Kratzwald et al. (2018)	Facebook 2895 posts	LSTM	1.007	3.519	-	MSE
		BiLSTM	0.990	3.550	-	
Akhtar et al. (2019)	Facebook 2895 posts	Multi-task learning	0.727	0.355	-	Pearson correlation
	Emobank		0.635	0.375	0.277	
Godinho (2018)	ANET	BiRNN + MP	0.706	0.299	-	Pearson correlation
			1.963	4.575	-	MAE
			4.777	25.008	-	MSE
	EmoBank	BiLSTM + MP + Attention	0.553	0.348	-	Pearson correlation
			0.268	0.251	-	MAE
			0.127	0.104	-	MSE
	Facebook	BiLSTM + MP + Attention	0.725	0.925	-	Pearson correlation
			0.695	0.613	-	MAE
			0.743	0.650	-	MSE
Buechel et al. (2018)	ANET	CNN	mean 0.70		Pearson correlation	
		LSTM	mean 0.73		Pearson correlation	
		GRU	mean 0.74		Pearson correlation	
	ANPST	CNN	mean 0.45		Pearson correlation	
		LSTM	mean 0.56		Pearson correlation	
		GRU	mean 0.57		Pearson correlation	
	MAS	CNN	mean 0.62		Pearson correlation	
		LSTM	mean 0.65		Pearson correlation	
		GRU	mean 0.69		Pearson correlation	

Table 2.2: Most important works in Assigning sentiment to Textual Utterances

2.3 Overview

This chapter started with a description of fundamental concepts, discussing how to encode textual information, giving a brief introduction to deep neural networks and deep learning, and explain how to represent emotions. the chapter also provided an overview of the work done in the field of automatic emotion extraction, for both words and textual utterances.

The next chapter describes the proposed models, using both simple heuristics and more complex machine learning algorithms. It will also describe how the textual information is represented to allow the use of cross-lingual mechanisms.

Quantify Emotion in Multiple Languages

To my knowledge, there is still a the need to better answer the question of how to infer emotions from text written in languages with few training resources. In order to answer this question, it was necessary to compare different types of neural networks (e.g., CNNs and LSTMs), to understand which of them perform better and why. It was also necessary to approach word norms and textual utterances separately.

This chapter starts with an explanation of textual representation methods (in Section 3.1) particularly focusing on the problem of a multilingual spaces. Then, Section 3.2 contains an explanation of all the models produced to quantify emotions from both words (in Subsection 3.2.1) and short texts (in Subsection 3.2.2). In the conclusions of this chapter, there is a couple of screenshots of the website that I produced to showcase

3.1 Text Representation in a Multilingual Space

It is necessary to understand that the machine learning algorithms may require textual information as input. However, the words, from a textual utterance, can not be given directly to a model. So, it is necessary to first encode the text in a way that is intelligible to a machine.

Word embeddings are vector representations for words, responsible for capturing their semantic or syntactic meaning. Several approaches were suggested over the years. Word2Vec (Mikolov et al., 2013b), also based on the skip-gram (Mikolov et al., 2011) model, is responsible for predicting the context of a given the word. However, the Word2Vec method does not allow the representation of words out of the vocabulary. FastText (Joulin et al., 2016), based on the skip-gram model (Mikolov et al., 2011), proposes the use of word fragments to express word vectors, allowing us to represent words out of the vocabulary.

In our models, we used FastText word vectors pre-trained on Common Crawl and Wikipedia, originally proposed by Grave et al. (2018). These embeddings are available in 157 different languages. For a more in depth explanation of the FastText representation, go to Section 2.1.1.

However, it was still necessary to represent word embeddings in a multi-language space.

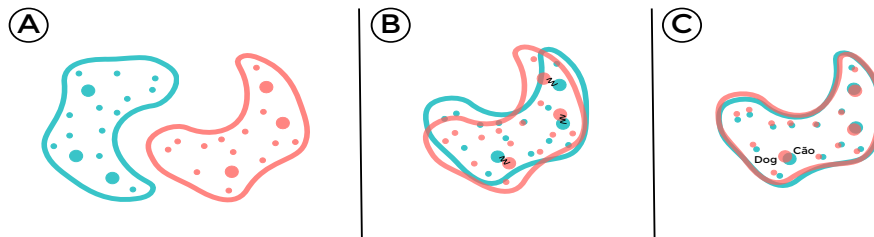


Figure 3.1: Cross-lingual alignment method from Conneau et al. (2017).

One of the state-of-the-art methods that produces bilingual FastText embeddings without any training data was proposed by Conneau et al. (2017). To produce these bilingual embedding space a two-step method (Faruqui and Dyer, 2014) was applied. After obtaining a monolingual word embedding for both languages separately, is necessary to transformer them into the same bilingual vector space by a linear transformation. This alignment is performed by selecting anchor points, from the most frequent words, and mapping the less frequent words through a distance metric, namely cross-domain similarity local scaling. Figure 3.1 illustrates these two steps, where A corresponds to a representation of two embeddings of different languages. In B, we learn a rotation matrix to adjust the embeddings of a language according to the other. In C, the mapping is improved by a method called Procrustes, that uses the most frequent words of the training corpus in a determined language (represented by the bigger sized dots). The MUSE dataset contains a mapping between English to 30 languages, including Portuguese.

An improvement to MUSE was made by Chen and Cardie (2018). The Unsupervised Multilingual Word Embeddings (UMWE) method is a multilingual generalisation of the MUSE approach. MUSE only considers the parity between two languages. For example, let us consider we want to translate both Portuguese and Spanish to English. If we translate them individually to English, we are neglecting the source language similarities, capturing therefore producing a worse multilingual embedding space. To solve this incapability of the dependencies between several languages, UMWE developed two steps: Multilingual Adversarial Training (MAT) and Multilingual Pseudo-Supervised Refinement (MPSR).

On the first step, it is necessary to resort to discriminators which are trained to detect if a vector corresponds to a determined language. In other words, all the source languages will be iterated, and two will be selected alongside a batch of words from each language. Each word will be encoded into the target space, and from the target space into the other language space. Then the loss function is calculated, determining if the generated embedding is a real embedding of the second language. Considering the result, the discriminators will be updated. This algorithm also enables us to translate a determined word to the target space and back to

itself, improving the performance of the model. The goal of doing these operations is to mix the discriminators, one for each language, and improving the performance by observing similarities between languages while training.

On the second step, the goal is to refine the embeddings obtained in the previous step. This is achieved by observing the most frequent words of each language, producing a pairwise dictionary, and improving the embeddings by pseudo supervised refinement. In other words, the algorithm this involves comparing a vocabulary of the most used words of the source languages with the ones on the target language (i.e., paring the words of the two languages considering a kNN technique), where words from each language will be encoded directly into the second language space. This way, the second step refining the mapping matrix of each language.

In my study, it was necessary to run the UMWE framework, with the target embedding defined as English and the source languages being other five languages (i.e., Portuguese, Spanish, German, Italian and Polish). I used the FastText embeddings extracted from the website Grave et al. (2018). The outputs generates mapping matrices between each language and the target language. So, whenever the models receive a text, first the text is divided into elements (i.e., words and punctuation). Then, each element is assigned an embedding, using the FastText embeddings. If the language of the data is not English, the embeddings are multiplied by the mapping matrices, responsible for aligning the original embedding into the English embedding space. Then, the text is ready for the ML models.

3.2 Proposed Models

The models described next were develop with the Python¹ programming language, alongside several libraries such as keras², FastText, numpy, Torch, and TensorFlow, among others. A framework was also used, UMWE (Chen and Cardie, 2018), for converting several language embeddings into one target embedding space through a translation matrix.

The training of the models was done using the optimizer named Adam. As we already discussed in Section 2.1.2, Adam is a type of gradient descent optimization algorithm, combining two well known optimizers: AdaGrad and RMSProp. This optimizer is usually used for problems involving large amounts of training data, since it sustains the learning rates. The optimizer reaches really good performances and is also faster, compared to other optimization algorithms.

¹<https://www.python.org>

²<https://keras.io>

The optimizer named AdaGrad was also tested but performed worst.

To allow a better comparison between models, it was necessary to standardize the use of some hyperparameters. The number of samples for the model to analyze, the batch size, was fixed to 64. The number of epochs, i.e. the number of times the model will go through the training data, was fixed to 200. This number should be large to allow the model to run enough times for the error to be minimized and prevent redundant alterations of the weights.

To facilitate the use of the datasets (i.e. some had values between 0-9, other between 0-5), I pre-processed the values of the emotional dimensions of the datasets, so that every emotion is associated to values between $[0,1]$. On the datasets that did not contemplate the dominance dimension, I opted to add a dominance column with the value "-1". Since this would severely affect the backpropagation, it was necessary to customize the loss function used to train the model, in order to ignore all the negative ground-truth values that were "-1".

The next sections describe how the models were designed to tackle the problems described in the introduction. Section 3.2.1 describes the models that infer emotion ratings of words, and Section 3.1 describes the models used to infer emotion ratings of textual utterances. All the adjustable parameters that required testing (e.g., to infer the best values), will be described in the next chapter.

3.2.1 Quantify Emotion from Words

One of the objectives of this work is to infer emotion rating from words, considering not only English but also other languages, leveraging a cross-lingual embedding space. The aforementioned word representations were used together with four different types of forecasting models, namely a kNN interpolation approach, random forest regression, kernel ridge regression, and a MLP. The four approaches were implemented through the scikit-learn (Pedregosa et al., 2011) and Keras libraries. They all apply to multi-output problems, where the same predictor variables are used to predict several outputs (i.e. in our case, valence, arousal and dominance scores are predicted simultaneously). All the following models receive as input a 1-dimensional numpy vector of size 300, which is the embeddings size of a word, calculated using the FastText method.

In the kNN interpolation approach, for each word in the test set, we identify the set of k most similar words (as measured according to the Euclidean distance between the word embeddings) in the training set, and assign the weighted mean rating of these words to the target word, as the extrapolated rating. The kNN are weighted such that nearby instances contribute more to the

final scores than faraway instances, namely by considering weights proportional to the inverse of the distance from the query instance. The value for k is an optimization parameter.

Random forests are ensemble ML algorithm, based on randomized decision trees (Breiman, 2001). This method is based on building a group of decision trees. In this algorithm, every tree is generated by a different subset of features from different drawn samples (i.e. with replacement from the entire training set, called bootstrap samples) of the full dataset. Using different features prevents the algorithm from overfitting the model. Each tree in the ensemble is built through the Classification and Regression Trees (CART) algorithm (Breiman et al., 1984). This algorithm constructs each binary decision tree in the ensemble using the feature and threshold that yield the lowest mean-squared error at each node. Given our multi-output setting (i.e., we simultaneously attempt to predict valence, arousal and dominance), the leaves of the trees store three output values, and the splitting criteria compute the average MAE across all three outputs. Alongside this, when a node is split, during the creation of the tree, the split that is chosen is the split with the best Gini index, among the random subset of features. The bias of the forest tends to grow lightly due to the randomness of the feature selection. Nonetheless, the variance also tends to decrease due to averaging, it generally it is compensated by the increase of the bias and therefore yielding overall better models.

The main parameters to adjust when using random forests correspond to the number of trees used in the forest (i.e. if the number is higher, it will obtain better results, but it will also take longer to compute) and the size of the random subsets of the features that should be regarded when splitting a node of the tree (i.e. the lower the greater the raise of the bias, but also the reduction of variance). An empirically right approach for the case of regression problems is to set the size of the random subset of features equal to the total number of features, three in this case. As for the number of trees, it was fixed at 200 in our experiments, since it was the best value for this parameter. The maximum depth of the tree was set to 50. To provide quicker training, the number of parallel jobs was set to -1, meaning that all processors will be used to train the model.

The kernel ridge regression approach combines the standard ridge regression (i.e. linear least squares with l2-norm regularization) with the kernel trick, as used in Support Vector Machines. This algorithm learns a linear function in the space produced by the kernel and the data, which for a non-linear data, translates into a non-linear function in the original space. Standard ridge coefficients minimize a penalized residual sum of squares, corresponding to:

$$\min_{\mathbf{w}} \|\mathbf{X} \cdot \mathbf{w} - \mathbf{y}\|_2^2 + \alpha \|\mathbf{w}\|_2^2 \quad (3.1)$$

In the previous formula, \mathbf{X} is the matrix of explanatory variables (i.e., the word embedding values), \mathbf{y} is a vector with the target values, and $\alpha \geq 0$ is a regularization parameter that controls the amount of shrinkage (i.e., the larger the value of α , the greater the amount of shrinkage, and thus the coefficients become more robust to co-linearity). In this thesis, given that I aim at predicting valence, arousal and dominance, I build independent ridge regression models, i.e. one for each of the three outputs. Kernel ridge regression extends the general setup considered above to allow for nonlinear prediction functions. For an arbitrary instance $\mathbf{x} \in \mathbb{R}^n$, the outcome suggested by ridge regression (i.e., $\mathbf{w}^\top \cdot \mathbf{x}$) can be rewritten into the dual form of the ridge regression solution (i.e., $\mathbf{w}^\top \cdot \mathbf{x} = \mathbf{y}^\top (\alpha \mathbf{I} + \mathbf{X}\mathbf{X}^\top)^{-1} \mathbf{X}\mathbf{x}$). When using the dual form, and because we only need scalar products between instances, we can directly use a kernel function to map instances into a higher-dimensional feature space, where regression can often be made more effectively. A popular choice for the kernel, which we used in our experiments, is a radial basis function of the form $k(\mathbf{x}_a, \mathbf{x}_b) = \exp(-\gamma |\mathbf{x}_a - \mathbf{x}_b|^2)$ with $\gamma > 0$. The values for α and γ are optimization parameters associated with the kernel ridge regression method.

Finally, the MLP mimics the synaptic connections between the neurons in our brain. This particular model is composed of three types of layers (i.e., input, hidden and output layers). The model learns by taking a set of inputs (x_1, x_2, \dots, x_n). The inputs pass through a set of neurons, each multiplying the input by weights, delivering a result through the output layer. All the functions for the MLP are described in Section 2.1.2.

The MLP model was built through Keras, an open-source library integrated on top of TensorFlow, to allow building deep learning models. In my specific model, we have one input layer with the size of the embedding vector (vector of 300 doubles, in this particular case). Then a hidden state, created by a fully connected layer, is composed of 100 neurons, plus the bias. In this layer, the weights (referred to as the *kernel_initializer* parameter) were initialized randomly and the biases with zeros. I also applied the ReLu activation function. To decrease the chance of overfitting, I considered set a weight regularizer (*kernel_regularizer* parameter) as the L2 norm with the value 0.0001. The L2 norm, also known as the Euclidean norm, calculates the shortest distance between two points by summing the squared weights. On the output layer, we have a fully connected with three neurons, one for each emotional dimension that we are considering. This layer has a linear activation function because of the continuous output values. This MLP was trained through 200 epochs, with a batch size of 64 and the Adam optimization

algorithm (Kingma and Ba, 2014) optimizer.

3.2.2 Quantify Emotion from Textual Utterances

This thesis proposes thirteen models, based on approaches described in the related work section. When the textual content is provided to the network, each word is lowercased and converted to the embedding of the proper language (including punctuation, since these symbols also provide usefully information for inferring emotions). If the words are not in English, each word embedding is multiplied by the translation matrix. Then, the utterances are ready as input.

In this section, we will start by dividing the models into two different categories: simple models exploring averages (Section 3.2.2.1) and models exploring machine learning (Section 3.2.2.2).

3.2.2.1 Simple Models Exploring Averages

One of the models that had a good performance when inferring emotion ratings for words was a simple MLP. This MLP was adapted and trained with datasets presenting affective norms for words from six different languages: English (Scott et al., 2019; Warriner et al., 2013; Bradley and Lang, 1999), Spanish (Redondo et al., 2007), Portuguese (Soares et al., 2012), Italian (Montefinese et al., 2014), German (Schmidtke et al., 2014), and Polish (Imbir, 2016a). In all the datasets that are not English, I also had access to a column in English where the original text was translated. In those cases, when training the model, we considered both the word in English and on the original language.

To observe the need for syntactic information when analyzing sentiment from the written, I created four models based on the MLP. These model do not take into consideration the syntactic structure of the textual utterances, and are models based on simple statistical approaches.

On the first model (Figure 3.2.a), each embedding is provided to the pre-trained MLP, giving the dimensions of each word. The dimensions are then be summed and an average is calculated. On the second model (Figure 3.2.b), I made an average of all the embedding and this average was then provided as input to the pre-trained MLP.

The last two models are more complex than the previous ones. On the third experiment, I performed an average pooling of the embeddings, considering windows of sizes between one and five words. The average of all these pooling windows was calculated, and finally provided to

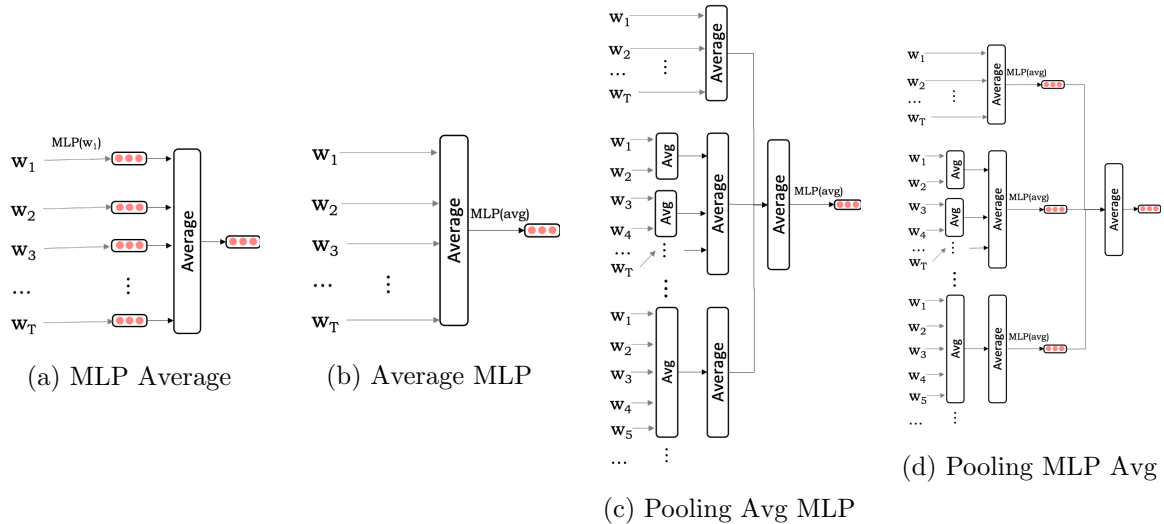


Figure 3.2: Models for Assigning Word-level Sentiment

the MLP. The last model corresponds to a small change, since the MLP was applied after each pooling window and I later calculated the average. Later, it was added a model similar to the two previous ones, however, using the MLP on each word and then doing the Pooling average. The models can be seen in Figure 3.2.

3.2.2.2 Models Exploring Machine Learning

It is first necessary to note that all the models described in this section can receive textual utterances with a maximum of 200 words. Since the models must be trained with a determined input vector size, if a determined input text has less than 200 words, all the remaining positions are filled with vectors of 300 zeros. For the zeros to be ignored in the models, all of them consider a masking layer (*RemoveMask*), responsible to remove all the zeros.

Yann LeCun, inspired by a model of the human visual cortex put forward by Hubel and Wiesel (1962), developed the Convolution and Pooling architecture (LeCun et al., 1995), also known as CNN. The first studies that applied CNNs to NLP were conducted by Collobert et al. (2011) in the area of semantic-role labelling, with subsequent studies by Kalchbrenner et al. (2014) and Kim (2014) respectively in the field of sentiment analysis and question-type classification. The main goal of CNN is to detect patterns across space, by firing when a determined pattern of words is compared to a determined filter. CNNs are composed of two layers, namely convolution and pooling.

Convolution Layer receives two inputs: a text translated into embeddings and a filter. The vector of embeddings is multiplied by the filter generating a Feature Map. Each filter takes into

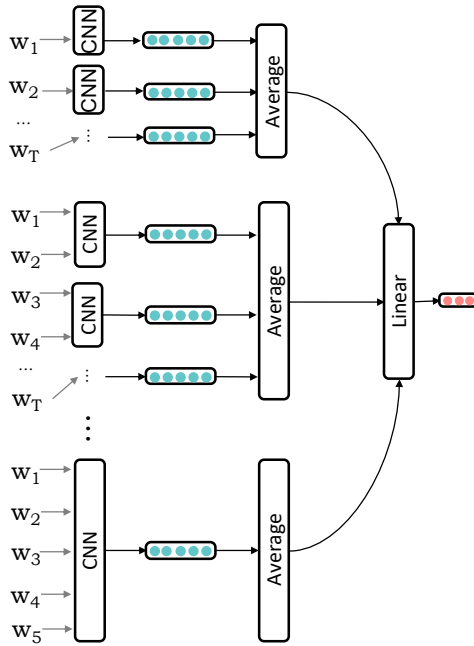


Figure 3.3: Convolution and Pooling operations applied to a sentence.

consideration a specific feature and can have differently sized, depending on what window size you want to consider. In our experiments, we regarded as sizes of word windows between one and five, as shown in Figure 3.3. To reduce the feature maps, we pass them through a pooling layer, responsible for reducing the dimensionality while at the same time recalling the important information. There are two types of pooling operations, and we can see in Figure 3.3 that the one used in my models was average pooling, responsible for returning the average of all values.

Considering the basic CNN model, we did four minor alterations. The first model is identical to the one shown in Figure 3.3. In the next three, we applied the MLP model. In the first, I passed the embeddings through the MLP model and the results were input to the convolution layer. In the second, I applied the MLP to the output of the convolution layer, afterwards applying the average pooling. In the third, I applied the MLP model in the end, after the linear operation.

Even though CNNs have fast performance, LSTM models are more successful when working with natural language (Yin et al., 2017), since LSTM models tend to take into consideration the input as sequences (i.e. text, time series).

The LSTM model used in my experiments is the same as explained in Section 2.1.2. To enhance the representation of each word in the sentence (Schuster and Paliwal, 1997), we choose to use a Bidirectional LSTM (BiLSTM). The idea is to have two LSTMs travelling through the sentence at the same time, one that encodes the sentence left to right and, separately, other

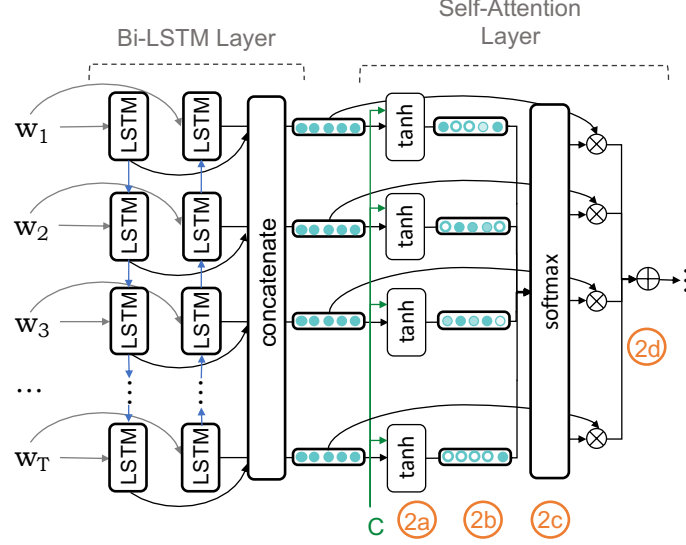


Figure 3.4: A model based on a BiLSTM and attention.

that travels from the end to the beginning of the sentence. In the end, we concatenate these two representations. This is shown as the BiLSTM layer of Figure 3.4.

As Yin et al. (2017) referred in their paper, tracing the whole sentence with an LSTM can disregard important keywords. So, aligned with the LSTM, I also used a self-attention layer, as shown in Figure 3.4 and expressed in Equation 3.2.

$$h_{i,j} = \tanh \left(x_i^\top W_1 + x_j^\top W_x + b_i \right) \quad (3.2a)$$

$$e_{i,j} = \sigma \left(W_a h_{i,j} + b_a \right) \quad (3.2b)$$

$$a_i = \text{softmax} \left(e_i \right) \quad (3.2c)$$

$$\text{self_attention}_i = \sum_j a_{i,j} x_j \quad (3.2d)$$

In self-attention, it is first necessary to calculate $h_{i,j}$ (Equation 3.2a) by summing the values of the current position and the previous, all previously multiplied by a weight matrix. Then, I multiply the values by the alignment weights, to get the alignment scores (Equation 3.2b). On Equation 3.2c, I apply softmax to the attention scores, for the values to vary between 0 and 1 and determine the probability of each given the word. At the end (Equation 3.2d), a_i corresponds to the amount of attention j^{th} should pay to the i^{th} input, and summing all the results.

First, I considered models with LSTM layers and a self-attention layer, as it can be visualized

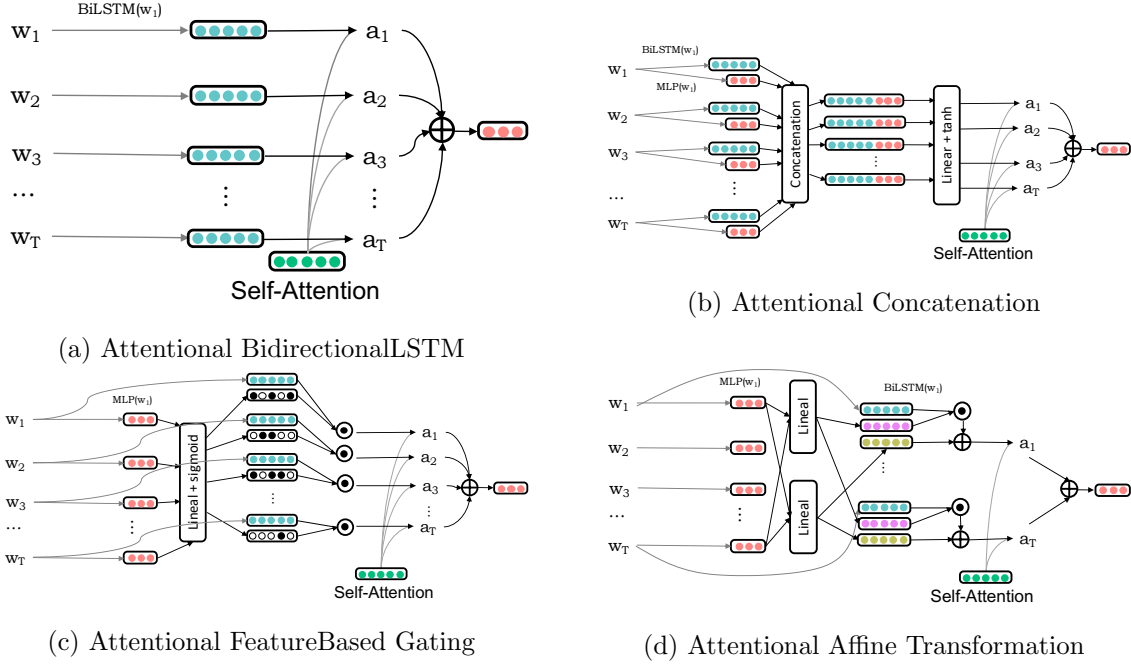


Figure 3.5: Proposed models applying Self-Attention and BiLSTM Layers.

in Figure 3.5a. I also generated an alteration to this model, where instead of receiving the embeddings as the input, the pre-trained MLP was applied to all the embeddings. The results of the operation were provided to the LSTM layer.

Three of the models were inspired by the work developed by Margatina et al. (2019), and they are here given the same names they had in the original paper.

The attentional concatenation model, shown in Figure 3.5b and Equation 3.3, calculates the BiLSTM representation of each embedding. In parallel, I applied the MLP pre-trained model for every word of the sentence. Then, I performed the concatenation of both results and pass that concatenation through a self-attention layer. In the end, the output is calculated through a feed-forward layer, with three dimensions, to predict the three emotional dimensions.

$$x_1 = \tanh (W_c [\text{BiLSTM} (w_1) \parallel \text{MLP} (w_i)] + b_c) \quad (3.3a)$$

operations 3.2a - 3.2d

$$d = l \cdot 3 + b \quad (3.3b)$$

The second method, described in Figure 3.5c and Equation 3.4, applies the MLP pre-trained model to the word embeddings and later uses linear plus sigmoid operations. Considering a gating mechanism, by applying the sigmoid function, the model will have a mask-vector where

each value varies between 0 and 1 that will later be applied to the embeddings of each word by an element-wise multiplication, represented by \odot . Lastly, we have the the self-attention layer.

$$f_g(h_i, \text{MLP}(w_i)) = \sigma(W_g \text{MLP}(w_i) + b_g) \odot h \quad (3.4)$$

In the final model a feature-wise affine transformation is applied, corresponding to a normalization layer preserving collinearity and ratios of distances. Primarily, the pre-trained MLP model was applied to the word embeddings, and I enforced a scaling and shifting vector to the results of the MLP. This model, initially inspired by Perez et al. (2018), allows one to capture dependencies between features by a simple multiplicative operation. The results of the linear operation γ over the MLP results are later multiplied element-wise with the results from the BiLSTM layer over the embeddings. Finally, these values were added to β , and we apply a self-attention layer, as shown in Equation 3.5d

$$f_a(h_1, \text{MLP}(w_i)) = \gamma(\text{MLP}(w_i)) \odot h_i + \beta(\text{MLP}(w_i)) \quad (3.5a)$$

$$\gamma(x) = W_\gamma x + b_\gamma \quad (3.5b)$$

$$\beta(x) = W_\beta x + b_\beta \quad (3.5c)$$

3.3 Overview

This chapter started with a description for how the textual contents were represented, followed by a report on how the main goals of my thesis were solved. Section 3.1 presented by models to infer sentiment from words, and Section 3.2 presented thirteen approaches to infer sentiment from textual utterances. The next chapter will describe the results of these models in datasets of five different languages.

4 Experimental Evaluation

This chapter explains all the processes behind the execution and validation of the models presented in the previous chapter. The chapter starts with a description of the datasets used in the tests considering both words (Section 4.1.1) and layer textual utterances (Section 4.1.2) this is followed by an explanation of the metrics used to validate the results of the models, with the results being presented in Section 4.3. In the conclusions of this chapter, there is a couple of screenshots of the website that I produced to showcase, by allowing the user to predict emotional ratings of a textual utterance with the the best models.

4.1 Datasets

This section is divided into two subsections. Section 4.1.1 describes the datasets used in the set of experiments to assign sentiment to words. In turn, Section 4.1.2 describes the datasets used to infer sentiment for layer textual utterances.

4.1.1 Word Affective Norms

The datasets used the set of experiments considering the assignment of emotion ratings to words are as follows:

- The Affective Norms for English Words (ANEW) (Bradley and Lang, 1999), composed of 1,034 unique words. This early work considering the three dimensions of valence, arousal, and dominance, with values between 1 and 9. Figure 4.1 suggests a homogeneous representation of words through the dimension space of valence and arousal, however, the dominance values are not that homogeneous.
- Warriner et al. (2013) extended the previous ANEW dataset, collecting 13,915 English lemmas and also including the three dimensions. For a richer dataset, data such as gender and education level of the individual providing the ratings was recorded, among others.

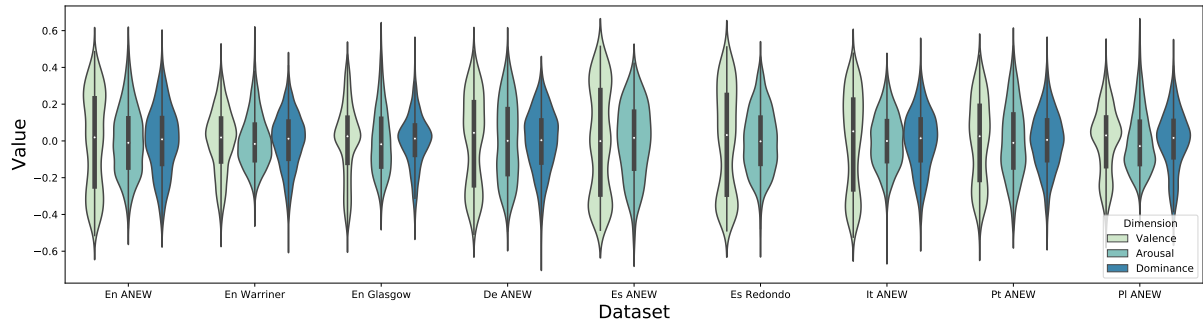


Figure 4.1: Comparison of the dimensional distribution of the words of the datasets in several languages.

- Scott et al. (2019) provided the Glasgow dataset with 5,553 English words with nine dimensions identified for each, including the three dimensions valence, arousal and dominance. This dataset features a worse spatial distribution, considering the two previous English datasets, of the words through the dimensions, as we can see on Figure 4.1.
- Schmidtke et al. (2014) (De ANEW) created an adaptation of the ANEW dataset. A total of 1,003 words were rated considering six dimensions (i.e. valence, arousal, dominance, arousal rated with a different metric, imageability and potency).
- Redondo et al. (2007) (Es Redondo) translated 1,034 Spanish words from the ANEW dataset and provided ratings based on 720 annotators, also considering into the three dimensions. Throughout the thesis, we will call this dataset the Redondo dataset.
- Stadthagen-Gonzalez et al. (2017) (Es ANEW) expanded the amount of Spanish emotional ratings by collecting ratings for 14,031 words. However, since the authors considered there was a strong correlation between valence and dominance, they chose to evaluate the words considering only valence and arousal. Throughout the thesis, the dataset will be called Spanish ANEW.
- Montefinese et al. (2014) (It ANEW) also translated all the words of the original English ANEW dataset, this time into Italian, and added some more words making a total of 1,121 Italian words. The annotators rated the words through the three dimensions, but also added psycholinguistic indexes. Homogeneous representation of valence.
- Soares et al. (2012) (Pt ANEW) provided an adaptation of the ANEW dataset for Portuguese. A total of 958 college students evaluated the translated words considering the three dimensions.
- Imbir (2016a) (Pl ANEW) also translated and extended the ANEW dataset to Polish.

Apart from the three dimensions, they also added a few parameters (i.e. importance, origin, concreteness, imageability, age of acquisition).

Through Figure 4.1 shows comparison of the datasets through a dimensional distribution of the words.

4.1.2 Text Affective Norms

In order to evaluate the models developed to extract emotional content from large textual utterances, I considered five datasets in three languages.

- The Facebook dataset, provided by Preoțiuc-Pietro et al. (2016), is composed by 2,895 English social media posts annotated considering two dimensions, i.e. valence and arousal.
- Bradley and Lang (2007) inspired by the ANEW dataset, created. The Affective Norms for English Tex (ANET) dataset, which is composed of 120 English texts, annotated considering the dimensions of pleasure (valence), arousal and dominance.
- Buechel and Hahn (2017) provided a dataset with 10,000 English sentences dataset, named Emobank. The sentences were annotated by an external annotator and by the writer of the sentences, considering the three dimensions VAD (i.e., valence, arousal and dominance) with values between 1 and 5.
- Pinheiro et al. (2017) created a European Portuguese dataset composed of 718 sentences. The sentences were rated considering the VAD dimensions, but also considering the six basic categorical emotions, with rating values ranging between 1 and 9.
- Imbir (2016b) created a dataset composed of 718 Polish sentences rated with the VAD dimensions, align with other parameters (i.e., origin, significance, and source).

Figure 4.2 it presents the dimensional distribution of the datasets, it is possible to inspect the homogeneity of the different dimensions.

4.2 Evaluation Metrics

All the results obtained from the experiments conducted in this thesis are evaluated both in terms of Pearson's correlation coefficient r and in terms of the Mean Absolute Error (MAE).

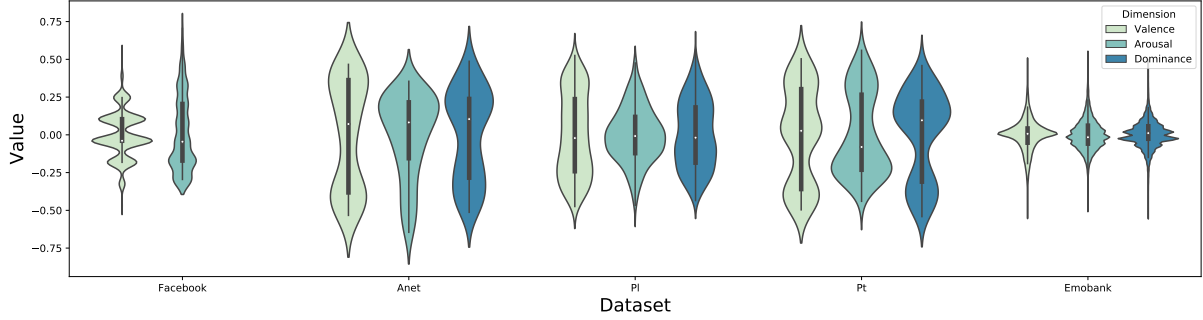


Figure 4.2: Comparison of the dimensional distribution of the datasets in several languages.

These metrics were chosen to compare this work with other studies conducted in this area (i.e. Preoțiuc-Pietro et al. (2016) or Akhtar et al. (2019), among others) and to better validate the produced models. These metrics can be computed as shown in the equations below, where x and y are sets with the obtained results and the ground truth measurements, respectively, and where $|e_i|$ is the absolute error for a testing instance i .

$$r(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x}) \times (y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \times \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (4.1)$$

$$\text{MAE}(x, y) = \frac{1}{n} \sum_{i=1}^n |x_i - y_i| = \frac{1}{n} \sum_{i=1}^n |e_i| \quad (4.2)$$

Another metric that is frequently used in the state-of-the-art, when assigning sentiment ratings to textual utterances is the MSE. This metric will also be considered on the experiments that focus on inferring sentiment from text when using ML models. This metric is computed using the following Equation:

$$\text{MSE}(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2 \quad (4.3)$$

4.3 Experimental Results

In this section, we will start by dividing into two different categories: assigning affective norms to words and assigning affective norms to textual utterances.

	ANEW Norm					
	Valence		Arousal		Dominance	
	Pearson	MAE	Pearson	MAE	Pearson	MAE
<i>k</i> -NN	0.830	1.029	0.606	0.820	0.682	0.585
Random Forest	0.767	1.251	0.547	0.939	0.647	0.632
Kernel Ridge	0.863	0.912	0.684	0.807	0.729	0.557
Multi Layer Perceptron	0.825	0.918	0.550	0.891	0.670	0.667
	Warriner Norm					
	Valence		Arousal		Dominance	
	Pearson	MAE	Pearson	MAE	Pearson	MAE
<i>k</i> -NN	0.858	0.914	0.615	0.640	0.761	0.669
Random Forest	0.793	1.146	0.573	0.720	0.727	0.737
Kernel Ridge	0.890	0.784	0.719	0.574	0.831	0.570
Multi Layer Perceptron	0.836	0.837	0.599	0.708	0.733	0.676
	Glasgow Norm					
	Valence		Arousal		Dominance	
	Pearson	MAE	Pearson	MAE	Pearson	MAE
<i>k</i> -NN	0.854	1.073	0.613	0.874	0.698	0.630
Random Forest	0.788	1.320	0.425	1.024	0.652	0.687
Kernel Ridge	0.892	0.922	0.592	0.894	0.751	0.579
Multi Layer Perceptron	0.838	0.905	0.515	0.951	0.665	0.702

Table 4.1: Obtained results when predicting ratings for words in the English ANEW, Warriner and Glasgow lexicons. The associated p -values for the Pearson product-moment correlation coefficient were always lower than 0.001.

4.3.1 Assigning Affective Norms to Words

In the first set of experiments including the usage of English words, I was considering a general monolingual approach. For this task, I selected the words in the set of norms from Warriner et al. (2013) and Scott et al. (2019) that did not appear in the ANEW corpus, as training data for predictive models that can later be used to estimate valence, arousal and dominance ratings for previously unseen words. The word embeddings were leveraged as features within different types of regression approaches, and I evaluated the obtained results in the task of predicting the valence, arousal and dominance ratings in ANEW. Table 4.1 presents the results obtained in our first set of experiments, considering two metrics (i.e. Pearson’s correlation and MAE).

The parameters associated with the k nearest neighbour, kernel ridge regression, and multilayer perceptron approaches were tuned through a simple grid-search, so as to optimize the average scores in all three emotional dimensions. By optimizing parameters according to the average correlation scores, I avoided over-fitting the models to individual cases. The best results were obtained for $k = 19$, $\alpha = 0.1$, $\gamma = 1$ and $max_{iter} = 250$.

A total of 12,764 words were used for model training, and evaluation was mostly made through the 1,026 words present in the ANEW lexicon. Nonetheless, I also present results when considering ratings for these same 1,026 words, as available in the Warriner and Glasgow datasets.

The results obtained with the four different types of prediction models are relatively similar, although the kernel ridge regression approach outperformed the others in terms of Pearson’s correlation and MAE. The correlation values are similar to those reported on the previous studies by Bestgen and Vincze (2012) and by Recchia and Louwerse (2015), even slightly superior. Comparing the best model presented in this thesis (i.e. 0.89 for valence, 0.72 for arousal, and 0.83 for dominance towards the rating Warriner, and 0.863, 0.684 and 0.729 towards ANEW), with the best models from Recchia and Louwerse (2015)(i.e. 0.80 , 0.62, and 0.66 towards Warriner and 0.74, 0.57, and 0.62 towards ANEW), it is possible to conclude that kernel ridge model even surpasses existing models.

For comparison, correlations between the valence, arousal and dominance ratings given in the original ANEW (Bradley and Lang, 1999) and the study by Warriner et al. (2013), respectively correspond to 0.95, 0.76 and 0.80. Considering the same correlation between ANEW and Glasgow, the values are 0.95 for valence, 0.66 for arousal, and 0.82 when considering dominance. In the study by Warriner et al. (2013), the authors report that typical correlations of human ratings across languages range from 0.85 to 0.97 for valence, 0.56 to 0.76 for arousal, and 0.77 to 0.83 for dominance. Correlations are somewhat lower between English speakers of different genders (i.e., 0.79, 0.52 and 0.59, for valence, arousal and dominance), different ages (i.e., correlations of 0.82, 0.50 and 0.59 when comparing subjects younger than 30 versus older than 30 years of age, respectively for valence, arousal and dominance), and different educational backgrounds (0.83, 0.47 and 0.61, respectively for valence, arousal and dominance), but remain large overall. Considering the Glasgow dataset, it is possible to observe almost the same correlations as in the previous dataset on gender (0.79, 0.52 and 0.59 for valence, arousal and dominance), age (0.82, 0.50 and 0.59 for the same order) and educational background (0.83 for valence, 0.461 for arousal and 0.61 for dominance). In general terms, the automatically estimated ratings obtained using the proposed method are at least as correlated with human ratings as male/female, old/young, and high/low education English speakers’ ratings are with each other, and in many cases even more so.

	<i>k</i> -NN			Random Forests			Kernel Ridge			Multi Layer Perceptron		
	Valence	Arousal	Dominance	Valence	Arousal	Dominance	Valence	Arousal	Dominance	Valence	Arousal	Dominance
ES	0.543	0.329	-	0.576	0.406	-	0.734	0.502	-	0.595	0.318	-
ES Redondo	0.789	0.546	0.704	0.736	0.537	0.670	0.845	0.656	0.784	0.789	0.524	0.668
PT	0.752	0.388	0.535	0.723	0.484	0.533	0.813	0.558	0.589	0.747	0.338	0.494
IT	0.762	0.466	0.621	0.710	0.504	0.590	0.831	0.600	0.700	0.759	0.428	0.594
DE	0.793	0.529	0.621	0.720	0.641	0.540	0.831	0.678	0.647	0.760	0.429	0.531
PL	0.312	0.283	0.196	0.437	0.424	0.280	0.566	0.487	0.366	0.351	0.261	0.215

Table 4.2: Pearson correlations obtained when predicting the ratings in four different adaptations of the ANEW lexicon, namely for the Spanish, Portuguese, Italian and German languages. The corresponding *p*-values were always lower than 0.001.

In the second set of experiments, we attempted to use information from the English language for extrapolating ratings to other languages, specifically Portuguese, Spanish, Italian, German and Polish. I leveraged adaptations of the original ANEW dataset into these four separate languages in order to evaluate the proposed approach (Redondo et al., 2007; Stadthagen-Gonzalez et al., 2017; Soares et al., 2012; Schmidtke et al., 2014; Montefinese et al., 2014; Imbir, 2016a). I again used representations for the English words in the set of norms from Warriner et al. (2013) and Scott et al. (2019), specifically for words that do not appear in the ANEW corpora for each target language, as training data for the predictive models. The representations for the English words are based on the same 300-dimensional skip-ngram FastText word embeddings that were pre-trained on Wikipedia, made initially available on FastText’s website.

In order to test the models, I extracted the FastText embedding of each word for each specific language (i.e., Portuguese, Spanish, Italian, German and Polish). However, to train predictive models that can later be used for extrapolating ratings to other languages, it was necessary to represent words in the target embedding space, the same used in the training data for the models. I then used UMWE Chen and Cardie (2018), an unsupervised approach for converting the source multilingual word embeddings. The UMWE framework is explained in Section 2.2.

Table 4.2 presents the results obtained in our second set of experiments. The parameters associated to the k nearest neighbour and kernel ridge regression approaches were again tuned through a simple grid-search, so as to optimize the average correlation scores in all three emotional dimensions, and across the four languages. The best averaged results obtained were almost the same, for $k = 19$, $\alpha = 0.1$ and $\gamma = 0.1$. The words used for model testing had to be present in each respective adaptation of the ANEW norms.

The obtained results show that relatively high correlations can be achieved for all five languages, although they are inferior to the results obtained for the monolingual setting. For comparison purposes, Table 4.3 shows the correlations between the norms for valence, arousal

	ANEW			Glasgow			Warriner et al.		
	Valence	Arousal	Dominance	Valence	Arousal	Dominance	Valence	Arousal	Dominance
Spanish	0.92	0.75	0.72	0.94	0.56	0.73	0.92	0.69	0.83
Spanish Redondo	0.96	0.80	-	0.95	0.63	-	0.92	0.69	-
Portuguese	0.91	0.58	0.62	0.91	0.30	0.60	0.90	0.57	0.67
Italian	0.92	0.63	0.75	0.93	0.43	0.75	0.92	0.62	0.75
German	0.90	0.64	0.60	0.92	0.30	0.63	0.91	0.66	0.69
Polish	0.87	0.65	0.69	0.87	0.47	0.67	0.84	0.64	0.56

Table 4.3: Correlations between human norms for English words and human norms in the four different adaptations of the ANEW lexicon. The corresponding p -values were always lower than 0.001.

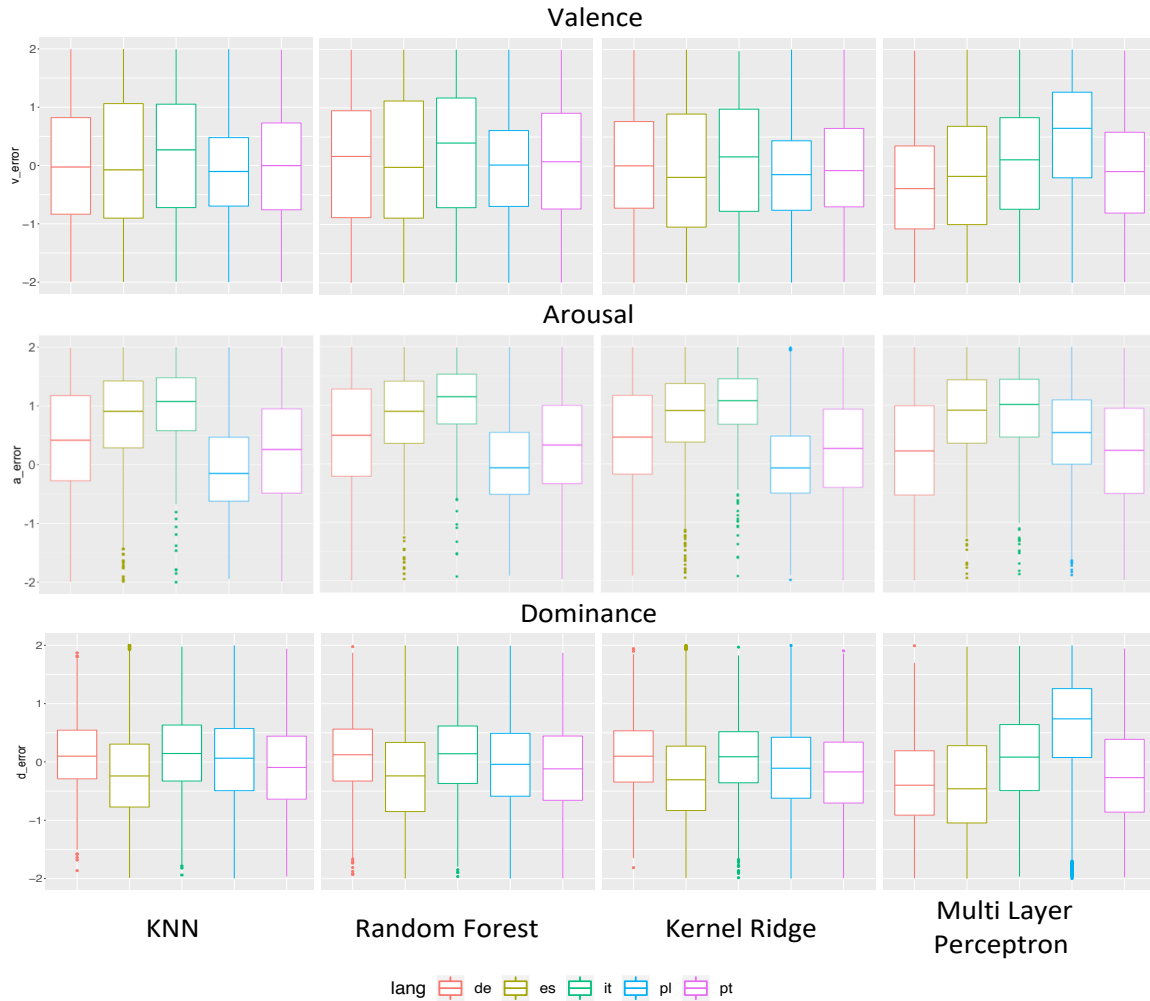


Figure 4.3: The absolute error between the affective norms predicted by the different models and the expected results.

and dominance, in the original ANEW dataset and in the set provided by Warriner and Glasgow datasets, against the norms in the five different adaptations of the ANEW dataset. It is interesting to notice that higher predictive accuracy is generally also obtained for the languages where the correlation towards the English norms is higher (i.e., Italian and Spanish).

Through Table 4.3, it is possible to observe how the different languages differed from the original ANEW dataset in terms of the correlations between human ratings. It is interesting to infer how this correlation affected the results, e.g., by also calculating the absolute error between the expected values and the results obtained. Figure 4.3 shows the obtained results for all five languages when tested with the four different models. The absolute error remains relatively stable when considering the valence dimension, and the error is higher on the arousal dimension, especially with Italian and Spanish languages. These results show how Latin derivative languages can transmit more enthusiasm than Germanic languages, thus confirming the statement made by Montefinese et al. (2014). In the same line of thought, Polish shows a lower correlation in

Table 4.3 as well as an underperformance on Table 4.2. A possible interpretation is the West Slavic origins of this language, conceiving a different meaning and affective norms of words from Germanic languages.

For comparison purposes, we also experimented with the training of kernel ridge regression models (i.e., the best performing method in the previous experiments) leveraging monolingual data (i.e., leveraging the skip-ngram embeddings trained separately for each of the four languages, together with the ANEW norms adapted to each of these languages), using a leave-one-out cross-validation methodology for evaluating the quality of the obtained results. The parameters k , α and γ were again kept at the same values considered for the experiments reported on Table 4.2. Table 4.4 presents the results from this particular experiment, showing that the obtained correlations are relatively similar to those obtained with the bilingual methodology. This finding further attests to the fact that the bilingual method can be a useful alternative to derive lexicons of emotion ratings for languages where no such norms exist, given that the resulting estimates will likely have a similar quality to those that would be obtained by extrapolating from small amounts of data in the target language.

4.3.2 Assigning Affective Norms to Textual Utterances

This section describes the experiments conducted to infer emotion rating from larger textual utterances. The section is divided into models exploring simple statistics and models exploring machine learning.

One of the models that had a better performance when inferring emotion ratings for words was the MLP. Hence, an MLP was pre-trained with all the datasets described in Subsection 4.1.1, and the result from this models were latter combined in order to try inferring emotion norms for larger pieces of text. Since one of the datasets (i.e., the Spanish Redondo dataset) does not have the dominance dimension, it was necessary to add a new column for dominance, filled

	Valence	Arousal	Dominance
Portuguese	0.731	0.520	0.507
Italian	0.821	0.625	0.681
Spanish ANEW	0.785	0.594	0.737
Spanish	0.748	0.675	-
German	0.720	0.699	0.637
Polish	0.627	0.675	0.748

Table 4.4: Obtained results, in terms of Pearson’s correlation coefficient, when using monolingual data through a leave-one-out cross validation methodology. The corresponding p -values were always below 0.001.

		Pt		Pl		Emobank		ANET		Fb	
		Pearson	MAE	Pearson	MAE	Pearson	MAE	Pearson	MAE	Pearson	MAE
MLP Average	V	0.686	0.234	0.499	0.227	0.359	0.086	0.639	0.301	0.384	0.154
	A	0.511	0.216	0.222	0.160	0.152	0.101	0.542	0.319	0.111	0.234
	D	0.470	0.238	0.312	0.187	0.058	0.093	0.261	0.263	-	-
Average MLP	V	0.625	0.232	0.429	0.226	0.284	0.073	0.697	0.312	0.298	0.132
	A	0.342	0.218	0.109	0.187	0.122	0.089	0.433	0.355	0.790	0.237
	D	0.579	0.234	0.436	0.194	0.123	0.122	0.622	0.258	-	-
Pooling	V	0.482	0.256	0.453	0.231	0.201	0.091	0.491	0.323	0.192	0.149
	A	0.187	0.231	0.166	0.160	0.110	0.122	0.420	0.316	0.79	0.250
	D	0.310	0.257	0.358	0.183	0.057	0.092	0.397	0.277	-	-
Pooling	V	0.537	0.249	0.456	0.231	0.224	0.094	0.492	0.323	0.193	0.148
	A	0.266	0.230	0.168	0.160	0.098	0.130	0.420	0.316	0.82	0.244
	D	0.405	0.263	0.359	0.183	0.068	0.100	0.396	0.360	-	-
Pooling	V	0.339	0.317	0.402	0.222	0.083	0.071	0.605	0.312	0.137	0.161
	A	0.330	0.253	0.335	0.188	0.029	0.088	0.515	0.336	0.152	0.208
	Avg	D	0.219	0.342	0.256	0.183	0.039	0.182	0.327	0.268	-

Table 4.5: Results obtained for statistical sentiment prediction of textual utterances, in terms of Pearson’s correlation coefficient and MAE.

with the value -1 and latter ignored through a custom loss function. Whenever the dominance dimension takes the value -1, the function will return zero, preventing the model to learn from those values.

4.3.2.1 Simple Models Exploring Averages

In a first experiment, the goal was to observe what were the models exploring simple statistics that had better performance, and compare them to more complex models. All the models that were tested in these experiments are described in Subsection 3.2.2.1.

The obtained results obtained are described on Table 4.5. Despite the simplicity of a model corresponding to an average (i.e., the MLP model is applied to each word of the text and an average of all the outputs is calculated to deliver a final output), this was the model that showed a better performance among the word-level solutions in almost every dataset.

4.3.2.2 Models Exploring Machine Learning

A second set of test explored machine learning models. To validate the models, it was necessary to conduct experiments using cross-validation.

Cross-validation is a simple method allowing us to validate a model (e.g. by calculating its precision) using all the available data, dividing a dataset into splits, usually between 2 and 5. A number of those splits are used to train the model, and the other is used to validate it, repeating the test with the multiple folds. Considering that my experiments use several datasets,

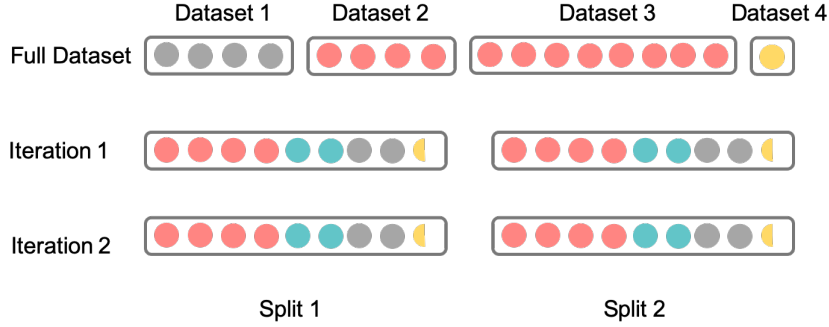


Figure 4.4: Example of cross-validation with multiple datasets.

		Pt			Pl			Emobank			ANET			Fb		
		Pearson	MAE	MSE	Pearson	MAE	MSE	Pearson	MAE	MSE	Pearson	MAE	MSE	Pearson	MAE	MSE
LSTM	V	0.641	0.184	0.059	0.507	0.184	0.055	0.536	0.070	0.009	0.769	0.207	0.059	0.547	0.100	0.018
	A	0.608	0.164	0.047	0.333	0.166	0.034	0.333	0.088	0.013	0.617	0.188	0.053	0.494	0.177	0.060
	D	0.576	0.164	0.056	0.445	0.149	0.042	0.092	0.120	0.065	0.439	0.231	0.082	-	-	-
MLP + LSTM	V	0.319	0.246	0.087	0.258	0.225	0.073	0.150	0.276	0.012	0.236	0.316	0.120	0.065	0.126	0.026
	A	0.232	0.241	0.071	0.108	0.146	0.034	0.016	0.297	0.013	0.254	0.282	0.097	0.126	0.235	0.081
	D	0.345	0.232	0.069	0.296	0.192	0.054	0.022	0.552	0.093	0.112	0.375	0.252	-	-	-
CNN	V	0.632	0.228	0.062	0.415	0.211	0.072	0.434	0.070	0.021	0.672	0.261	0.092	0.495	0.102	0.020
	A	0.312	0.241	0.050	0.241	0.148	0.059	0.170	0.087	0.032	0.493	0.221	0.168	0.260	0.212	0.058
	D	0.427	0.234	0.063	0.247	0.235	0.109	0.040	0.258	0.075	0.261	0.329	0.091	-	-	-
MLP + CNN	V	0.584	0.236	0.076	0.397	0.212	0.067	0.466	0.069	0.009	0.657	0.249	0.087	0.501	0.109	0.019
	A	0.345	0.221	0.063	0.281	0.146	0.034	0.136	0.089	0.013	0.536	0.204	0.060	0.316	0.215	0.065
	D	0.419	0.227	0.078	0.282	0.202	0.067	0.040	0.251	0.085	0.167	0.410	0.302	-	-	-
CNN + MLP	V	0.552	0.223	0.066	0.395	0.215	0.066	0.449	0.071	0.009	0.523	0.080	0.066	0.485	0.107	0.019
	A	0.343	0.219	0.034	0.197	0.145	0.034	0.214	0.088	0.013	0.393	0.114	0.058	0.315	0.215	0.067
	D	0.342	0.227	0.061	0.243	0.147	0.061	0.066	0.182	0.076	0.408	0.291	0.149	-	-	-
Attention Concat	V	0.691	0.177	0.057	0.435	0.202	0.064	0.507	0.073	0.010	0.649	0.238	0.007	0.561	0.101	0.019
	A	0.620	0.165	0.046	0.297	0.144	0.035	0.302	0.089	0.014	0.481	0.209	0.051	0.565	0.176	0.052
	D	0.663	0.167	0.049	0.348	0.182	0.050	0.363	0.122	0.074	0.283	0.276	0.004	-	-	-
Attention Feat Based	V	0.641	0.184	0.050	0.501	0.192	0.059	0.531	0.069	0.001	0.680	0.226	0.056	0.557	0.098	0.021
	A	0.608	0.164	0.042	0.391	0.137	0.031	0.320	0.083	0.014	0.538	0.198	0.051	0.545	0.174	0.057
	D	0.576	0.173	0.058	0.470	0.160	0.043	0.082	0.116	0.065	0.479	0.217	0.084	-	-	-
Attention Affine Transformation	V	0.569	0.206	0.072	0.434	0.206	0.065	0.501	0.074	0.010	0.728	0.225	0.070	0.523	0.108	0.057
	A	0.540	0.177	0.050	0.268	0.148	0.036	0.270	0.092	0.015	0.608	0.189	0.056	0.491	0.184	0.436
	D	0.473	0.218	0.075	0.338	0.180	0.051	0.075	0.143	0.067	0.481	0.266	0.119	-	-	-

Table 4.6: The prediction of valence, arousal and dominance with several models. The training and testing data are textual utterances from datasets in English, Polish and Portuguese.

it was necessary to divide each dataset equally between the splits. The process of using cross-validation with multiple datasets is shown in Figure 4.4, where each colour corresponds to a different dataset.

For this experiment, and considering the amount of time required to train each model (i.e. considering 200 epochs), I choose to divide the datasets between two splits. In the end, each split had the same amount of each dataset. Table 4.6 displays the results for each model in each dataset, considering Pearson’s correlation, MAE, and the MSE.

Through Table 4.6, it is possible to draw the following conclusions:

- Comparing the simple LSTM and CNN models, the LSTM shows a better performance in every dataset.

- It is possible to observe the variance between the values of the LSTM with and without the MLP layer of weights. It was expected that a pre-trained MLP layer would help to provide better predictions. However, by comparing the Pearson correlation on both tests, it is possible to observe worse results when using the MLP layer.
- The dimension that was more difficult to tackle was arousal, especially in the Facebook dataset.
- All the last three models had an overall better performance, compared to the rest of the models. Even though in some datasets (i.e., the Portuguese dataset) the results were similar to the simple average model (i.e., see the results in Table 4.5), it is possible to see an improvement on bigger datasets, such as the Facebook dataset.

From the work of Kratzwald et al. (2018), with a BiLSTM model, it is possible to observe an MSE correlation on the Facebook dataset of 0.990 and 3.550, respectively for valence and arousal. Comparing to the results obtained with the Attention Feature Based model for the Facebook dataset, it is possible to conclude that my proposal was able to outperform their results.

Considering the results obtained by Akhtar et al. (2019) (i.e., with a Pearson correlation of 0.727 and 0.355 for the Facebook dataset, and 0.635 and 0.375 for the Emobank dataset, respectively for valence and arousal), it is possible to observe that the Attention Concat model performed comparably (with my approach even showing better values for the dimension arousal than the work from Akhtar et al. (2019)). Ultimately, it is possible to conclude that the Attention Concat model has an overall good performance, even compared to models that were trained for one language.

The results obtained by a state-of-the-art study conducted by Godinho (2018) the a Pearson correlation of 0.553 and 0.348 for the Emobank dataset, and 0.725 and 0.925 for the Facebook data (i.e., for the dimensions valence and arousal, respectively). The results were obtained using a model composed by Bi-LSTM+Attention layers, and they are similar to my results using the Attention Feature Based model, that obtained results of 0.531 and 0.320 for Emobank, and 0.557 and 0.545 for Facebook. Comparing the results and considering that my model performed a little lower, although being trained with several idioms, the lower performance can be justified. It is possible to also see an improvement in my model regarding the MAE and MSE values, where for Emobank obtained 0.069 and 0.003 (i.e. MAE and MSE, respectively for the valence dimension), 0.083 and 0.013 (i.e. MAE and MSE, respectively for the arousal dimension): In turn, Godinho

(2018) reported 0.268 and 0.127 (i.e. MAE and MSE, respectively, for the valence dimension), 0.251 and 0.104 (i.e. MAE and MSE, respectively, for the arousal dimension).

In conclusion, the models presented in this thesis performed better than most state-of-the-art approaches, even considering that these models are not trained to tackle only one language. However, it would be interesting to compare this work considering datasets in even more languages.

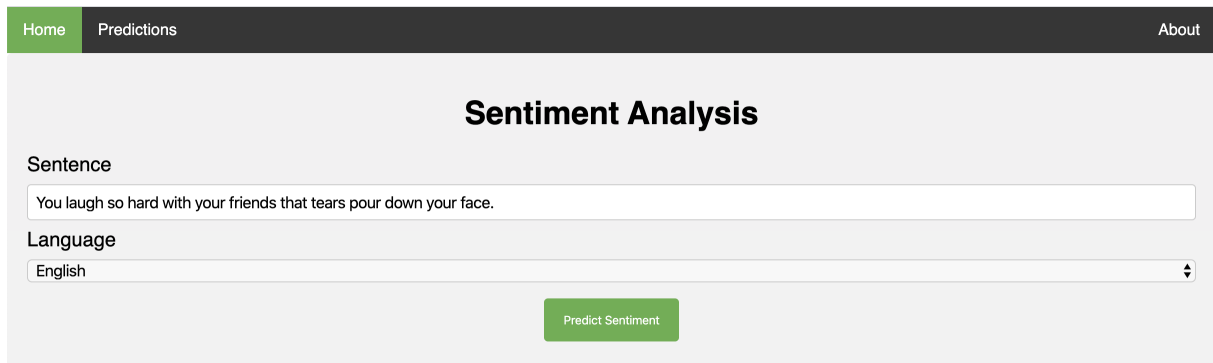
4.4 Overview

In this chapter, I presented the results obtained by executing and training the models presented in the previous chapter.

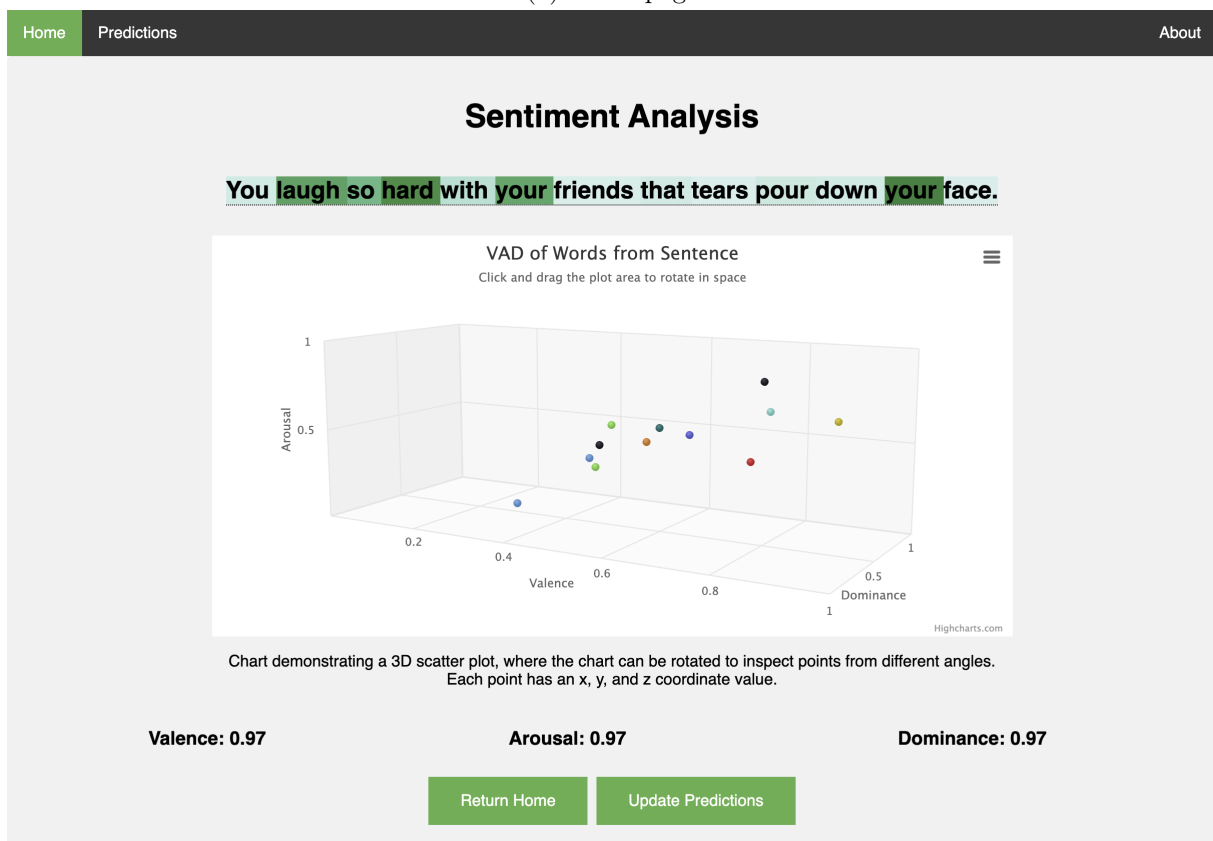
First, in Section 4.1, all the datasets in several languages that were used to validate the models. Then, in Section 4.2, I described the metrics used to validate and compare the results (i.e., metrics such as Pearson’s correlation, MAE and MSE).

Section 4.3 describes the experiments, results and conclusions drawn when trying to assign emotion ratings to both words and larger textual utterances. Section 4.3.1 showed that both the kernel ridge model and MLPs show similar correlation values compared to previous studies, even slightly superior. In Section 4.3.2 we first validate how is the performance of models that assign sentiment to text exploring a word-level statistics. Despite the simplicity of a model averages, the results show that this model outperformed the other tested models in almost every dataset. In Section 4.3.2.2 I describe the results for models that require training and consider the textual structure of the sentences. The models based on attention outperformed the rest of the models, and even compare to other state-of-the-art studies.

Considering the models tested on this chapter, I also developed a website that showcases the results, and through which a user can predict sentiment using the model that presented the best results overall: Attention Concat. In this website, it is possible to insert a sentence in a given language. When the prediction is calculated, the site shows the results aligned with the relevance that each word had in the prediction (i.e., the weights obtained through the self-attention for each word). The website also presents a 3D graph showing the results in the VAD dimensions for each separate word (i.e. using the pre-trained MLP model). Figure 4.5 presents screenshots of the site designed in the scope of this master thesis.



(a) Home page



(b) Prediction of a sentence

Figure 4.5: Website design in a prototype for predicting sentiment associated to input sentences.

Conclusions and Future Work



This research aimed to understand if it is possible for a machine learning model to quantify emotion expressed in text, in terms of valence, arousal and dominance, in multiple languages. To better answer the main research question, this study was divided into two parts (i.e., experiments concerning words and textual utterances).

5.1 Main Results

In a first set of experiments regarding the assignment of emotion ratings to words, I wanted to observe how well a ML model predicts sentiment in a monolingual scenario. To answer this question, it was necessary to train models (i.e. kNN, Random Forest, Kernel Ridge and MLP) with lexicons from Warriner and Glasgow datasets that did not appear in the ANEW corpus and tested with the words that appear in ANEW in the three corpora. The results showed promising results with the Kernel Ridge and MLP models. They show that a ML model can predict outcomes comparably to human annotators. It was also a goal to understand if ML model, trained with only English lexicons, could predict sentiment in other languages (i.e. Spanish, European Portuguese, Italian, German and Polish). The trained models used in the previous experience were used, and the results show that relatively high correlations can be achieved for all five languages. However, they are inferior to the results obtained for the monolingual setting. It was also interesting to notice that higher predictive accuracy is generally also obtained for the languages where the correlation towards the English norms is higher (i.e., Italian and Spanish).

Now, to understand if a ML model can quantify emotion in textual utterances of multiple languages, it was necessary to set the following secondary questions. What method provides better results: a word or text-level sentiment prediction for text? Are CNN's or LSTM's better for sentiment prediction? Do models with pre-trained MLP perform better or worst? What are the models that perform better in this scenario?

To answer the secondary questions, several models were created. An MLP was pre-trained with lexicons from six different languages. To infer the need to access all the syntactic structure

to infer sentiment from a text, four models that do not take into consideration the syntactic structure and do not require training were created. Furthermore, eight trainable models were conceived (i.e. LSTM, MLP+LSTM, CNN, MLP+CNN, CNN+MLP, Attention Concat, Attention Feature Based, Attention Affine Transformation), validated with two-fold cross-validation.

The results show that three trained models performed better (Attention Concat, Attention Feature Based, Attention Affine Transformation); however, the Average word-level prediction model also showed promising results. LSTM's tend to perform slightly better than CNN models. The difference was more evident in the arousal dimension. However, when the CNN and LSTM model were aligned with the pre-trained MLP, the results decreased, showing that a pre-trained MLP can decrease the performance of the model.

Overall, this thesis shows promising results when inferring sentiment, even in several languages. The main contribution of this work relies, first on the significant amount of models that were validated to infer how to extract sentiment from both words and textual utterances. There are few works on sentiment quantification, in particular, considering the dimensional way of quantifying sentiment. This thesis provides three trained models and one word-level model that show promising results compared to the state-of-the-art.

5.2 Future Work

For future work, it could be interesting to extend the experiments reported in this dissertation for word-level prediction of emotion ratings, considering also other languages and other types of lexical norms (e.g., leveraging data from the Bristol norms for age of acquisition, imageability, and familiarity), other types of forecasting models (e.g., different types of ensemble approaches, combining different modelling alternatives and choosing the best combination through cross-validation). It would also be interesting to test as the combination of skip-ngram word embeddings with other types of features, such as the incorporation of features based on word frequency, word length or orthographic similarity.

Besides fasttext embeddings, there are other distributional word representations that could also have been used in these thesis tests for comparison. Recent studies suggest that, after careful hyper-parameter tuning, there are no global advantages in any of the proposals from the recent literature. Still, for future work, it could be also interesting to experiment with word embeddings trained on different types of corpora (e.g., on social media data, that is perhaps more reflective of people's attitudes and emotions) and/or relying on different approaches, such

as the GloVe method.

On what regards predictions for larger pieces of text, it would be interesting to apply contextual encoders as word embeddings (i.e., Bert). It might also be interesting to apply different models to predict emotion ratings, such as models that use transformers.

Bibliography

- Akhtar, S., Ghosal, D., Ekbal, A., Bhattacharyya, P., and Kurohashi, S. (2019). All-in-one: Emotion, sentiment and intensity prediction using a multi-task ensemble framework. *IEEE Transactions on Affective Computing*.
- Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the International Conference on Learning Representations*.
- Banea, C. (2013). *Extrapolating Subjectivity Research to Other Languages*. PhD thesis, University of North Texas.
- Banea, C., Mihalcea, R., and Wiebe, J. (2013). Porting multilingual subjectivity resources across languages. *IEEE Transactions on Affective Computing*, 4(2).
- Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.
- Bebis, G. and Georgiopoulos, M. (1994). Feed-forward neural networks. *IEEE Potentials*, 13(4).
- Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3.
- Bestgen, Y. and Vincze, N. (2012). Checking and bootstrapping lexical norms by means of word similarity indexes. *Behavior Research Methods*, 44(4).
- Binali, H., Wu, C., and Potdar, V. (2010). Computational approaches for emotion detection in text. In *Proceedings of the IEEE International Conference on Digital Ecosystems and Technologies*.
- Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(1).
- Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

- Bradley, M. M. and Lang, P. J. (1999). Affective norms for english words (anew): Instruction manual and affective ratings. Technical report, The Center for Research in Psychophysiology, University of Florida.
- Bradley, M. M. and Lang, P. J. (2007). Affective norms for english text (anet): Affective ratings of text and instruction manual. *Technical Report. D-1, University of Florida, Gainesville, FL.*
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1).
- Breiman, L., Friedman, J., Stone, C., and Olshen, R. (1984). *Classification and Regression Trees*. The Wadsworth and Brooks-Cole Statistics-Probability Series. Taylor & Francis.
- Buechel, S. and Hahn, U. (2016). Emotion analysis as a regression problem—dimensional models and their implications on emotion representation and metrical evaluation. In *Proceedings of the European Conference on Artificial Intelligence*.
- Buechel, S. and Hahn, U. (2017). Emobank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*.
- Buechel, S. and Hahn, U. (2018). Word Emotion Induction for Multiple Languages as a Deep Multi-Task Learning Problem. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.
- Buechel, S., Sedoc, J., Schwartz, H. A., and Ungar, L. (2018). Learning neural emotion analysis from 100 observations: The surprising effectiveness of pre-trained word representations. *arXiv preprint arXiv:1810.10949*.
- Bullinaria, J. and Levy, J. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3).
- Calvo, R. A. and Mac Kim, S. (2013). Emotions in text: dimensional and categorical models. *Computational Intelligence*, 29(3):527–543.
- Chaudhari, S., Polatkan, G., Ramanath, R., and Mithal, V. (2019). An attentive survey of attention models. *arXiv preprint arXiv:1904.02874*.
- Chen, X. and Cardie, C. (2018). Unsupervised multilingual word embeddings. *arXiv preprint arXiv:1808.08933*.
- Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

- Citron, F. M., Weekes, B. S., and Ferstl, E. C. (2014). How are affective word ratings related to lexicosemantic properties? evidence from the sussex affective word list. *Applied Psycholinguistics*, 35(2).
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12.
- Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jégou, H. (2017). Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.
- Covington, P., Adams, J., and Sargin, E. (2016). Deep neural networks for youtube recommendations. In *Proceedings of the ACM Conference on Recommender Systems*.
- Dos Santos, C. and Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the International Conference on Computational Linguistics*.
- Ekman, P. (1992). An argument for basic emotions. *Cognition & Emotion*, 6(3-4).
- Ekman, P. and Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17(2).
- Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.
- Faruqui, M. and Dyer, C. (2014). Improving vector space word representations using multilingual correlation. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*.
- Godinho, J. D. F. (2018). *Extraction, Attribution, and Classification of Quotations in Newspaper Articles*. PhD thesis, University of Lisbon.
- Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57.
- Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation*.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8).

- Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *The Journal of Physiology*, 160(1):106–154.
- Hutto, C. J. and Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*.
- Imbir, K. K. (2016a). Affective norms for 4900 polish words reload (anpw.r): assessments for valence, arousal, dominance, origin, significance, concreteness, imageability and, age of acquisition. *Frontiers in psychology*, 7.
- Imbir, K. K. (2016b). Affective norms for 718 polish short texts (anpst): dataset with affective ratings for valence, arousal, dominance, origin, subjective significance and source dimensions. *Frontiers in psychology*, 7.
- Jasmin, D. and Casasanto, D. (2012). The QWERTY effect: How typing shapes the meanings of words. *Psychonomic Bulletin & Review*, 19(3).
- Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.
- Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.
- Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. M. (2017). OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.
- Köper, M. and Im Walde, S. S. (2016). Automatically generated affective norms of abstractness, arousal, imageability and valence for 350 000 german lemmas. In *Proceeding of the Conference on language Resources and Evaluation*.
- Köper, M., Kim, E., and Klinger, R. (2017). Ims at emoint-2017: emotion intensity prediction with affective norms, automatically extended resources and deep learning. In *Proceedings*

of the Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis.

Kratzwald, B., Ilic, S., Kraus, M., Feuerriegel, S., and Prendinger, H. (2018). Decision support with text-based emotion recognition: Deep learning for affective computing. *arXiv preprint arXiv:1803.06397*.

Kuperman, V., Stadthagen-Gonzalez, H., and Brysbaert, M. (2012). Age-of-acquisition ratings for 30,000 english words. *Behavior Research Methods*, 44(4).

LaBrie, R. C. and Louis, R. D. S. (2003). Information Retrieval from Knowledge Management Systems: Using Knowledge Hierarchies to Overcome Keyword Limitations. In *Proceedings off the Americas Conference on Information Systems*.

Lang, P. (1980). Self-assessment manikin. *Gainesville, FL: The Center for Research in Psychophysiology, University of Florida*.

LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, 3361(10).

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. (2006). A tutorial on energy-based learning. *Predicting Structured Data*, 1(0).

Li, M., Lu, Q., Long, Y., and Gui, L. (2017). Inferring affective meanings of words from word embedding. *IEEE Transactions on Affective Computing*.

Lund, K. and Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2).

Ma, C., Prendinger, H., and Ishizuka, M. (2005). Emotion Estimation and Reasoning Based on Affective Textual Interaction. In *Proceedings of the Affective Computing and Intelligent Interaction, First International Conference*.

Malheiro, R., Oliveira, H. G., Gomes, P., and Paiva, R. P. (2016). Keyword-based Approach for Lyrics Emotion Variation Detection. In *Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, pages 33–44.

Mandera, P., Keuleers, E., and Brysbaert, M. (2015). How useful are corpus-based methods for extrapolating psycholinguistic variables? *The Quarterly Journal of Experimental Psychology*, 68(8).

- Margatina, K., Baziotis, C., and Potamianos, A. (2019). Attention-based Conditioning Methods for External Knowledge Integration. *arXiv preprint arXiv:1906.03674*.
- McCann, B., Bradbury, J., Xiong, C., and Socher, R. (2017). Learned in Translation: Contextualized Word Vectors. In *Proceedings of the Annual Conference on Neural Information Processing Systems 30*.
- McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4).
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Mikolov, T., Deoras, A., Povey, D., Burget, L., and Cernocký, J. (2011). Strategies for training large scale neural network language models. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition & Understanding*,, pages 196–201.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proceedings of the Annual Conference on Neural Information Processing Systems*.
- Minsky, M. and Papert, S. (1969). Perceptron expanded edition.
- Montefinese, M., Ambrosini, E., Fairfield, B., and Mammarella, N. (2014). The adaptation of the affective norms for english words (anew) for italian. *Behavior research methods*, 46(3).
- Moors, A., De Houwer, J., Hermans, D., Wanmaker, S., Van Schie, K., Van Harmelen, A.-L., De Schryver, M., De Winne, J., and Brysbaert, M. (2013). Norms of valence, arousal, dominance, and age of acquisition for 4,300 dutch words. *Behavior research methods*, 45(1):169–177.
- Ng, A. Y., Jordan, M. I., and Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In *Proceeding of the Annual Conference on Neural Information Processing Systems*.
- Ng, H. T., Goh, W. B., and Low, K. L. (1997). Feature selection, perceptron learning, and a usability case study for text categorization. *ACM SIGIR Forum*, 31(SI).
- Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *Proceedings of the International Conference on Machine Learning*.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher,

- M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(1).
- Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. (2018). Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Perugini, M. and Bagozzi, R. P. (2001). The role of desires and anticipated emotions in goal-directed behaviours: Broadening and deepening the theory of planned behaviour. *British Journal of Social Psychology*, 40(1).
- Pinheiro, A. P., Dias, M., Pedrosa, J., and Soares, A. P. (2017). Minho affective sentences (mas): probing the roles of sex, mood, and empathy in affective ratings of verbal stimuli. *Behavior research methods*, 49(2).
- Preoțiuc-Pietro, D., Schwartz, H. A., Park, G., Eichstaedt, J., Kern, M., Ungar, L., and Shulman, E. (2016). Modelling valence and arousal in facebook posts. In *Proceedings of the Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*.
- Recchia, G. and Louwerse, M. M. (2015). Reproducing affective norms with lexical co-occurrence statistics: Predicting valence, arousal, and dominance. *The Quarterly Journal of Experimental Psychology*, 68(8).
- Redondo, J., Fraga, I., Padrón, I., and Comesaña, M. (2007). The spanish adaptation of anew (affective norms for english words). *Behavior research methods*, 39(3).
- Roy, N. and McCallum, A. (2001). Toward optimal active learning through monte carlo estimation of error reduction. *Proceeding of the International Conference on Machine Learning*.
- Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.
- Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6).
- Russell, J. A. and Mehrabian, A. (1977). Evidence for a three-factor theory of emotions. *Journal of research in Personality*, 11(3).
- Schmidtke, D. S., Schröder, T., Jacobs, A. M., and Conrad, M. (2014). Angst: Affective norms for german sentiment terms, derived from the affective norms for english words. *Behavior research methods*, 46(4).

- Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11).
- Scott, G. G., Keitel, A., Becirspahic, M., Yao, B., and Sereno, S. C. (2019). The glasgow norms: Ratings of 5,500 words on nine scales. *Behavior research methods*, 51(3).
- Sedoc, J., Gallier, J., Ungar, L., and Foster, D. (2016). Semantic word clusters using signed normalized graph cuts. *arXiv preprint arXiv:1601.05403*.
- Sedoc, J., Preoțiu-Pietro, D., and Ungar, L. (2017). Predicting emotional word ratings using distributional representations and signed clustering. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*.
- Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., and Zhang, C. (2018). Disan: Directional self-attention network for rnn/cnn-free language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Soares, A. P., Comesaña, M., Pinheiro, A. P., Simões, A., and Frade, C. S. (2012). The adaptation of the affective norms for english words (anew) for european portuguese. *Behavior research methods*, 44(1).
- Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
- Specht, D. F. (1990). Probabilistic neural networks. *Neural networks*, 3(1).
- Stadthagen-Gonzalez, H., Imbault, C., Sánchez, M. A. P., and Brysbaert, M. (2017). Norms of valence and arousal for 14,031 spanish words. *Behavior research methods*, 49(1):111–123.
- Tang, D., Wei, F., Qin, B., Zhou, M., and Liu, T. (2014). Building large-scale twitter-specific sentiment lexicon : A representation learning approach. In *Proceedings of the International Conference on Computational Linguistics*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is All you Need. In *Proceedings of the Annual Conference on Neural Information Processing Systems*.
- Wang, J., Yu, L.-C., Lai, K. R., and Zhang, X. (2016). Community-based weighted graph model for valence-arousal prediction of affective words. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11).

- Warriner, A. B., Kuperman, V., and Brysbaert, M. (2013). Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior Research Methods*, 45(4).
- Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*.
- Yin, W., Kann, K., Yu, M., and Schütze, H. (2017). Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*.
- Yu, L.-C., Lee, L.-H., Hao, S., Wang, J., He, Y., Hu, J., Lai, K. R., and Zhang, X. (2016). Building chinese affective resources in valence-arousal dimensions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*.
- Zahiri, S. M. and Choi, J. D. (2017). Emotion Detection on TV Show Transcripts with Sequence-based Convolutional Neural Networks. *arXiv preprint arXiv:1708.04299*.
- Årup Nielsen, F. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC Workshop on 'Making Sense of Microposts': Big things come in small packages*.