Leibniz Universität Hannover

L3S

Master Thesis

Multilingual and Cross-Modal Embedded Representation for Tweets Aggregation

Author:	Houcem Ben Makhlouf
Matriculation Number:	10030607
Address:	Hiltenspergerstr. 77
	80796 München
Jury Members:	Prof. Dr. techn. Wolfgang Nejdl
	Prof. Dr. Ralph Ewerth
Advisors:	Dr. Erick Elejalde
	Sergej Wildemann, M.Sc.
Begin:	22.02.2022
End:	04.10.2022

I, Houcem Ben Makhlouf, declare that this master's thesis, and the work indicated herein have been composed by myself, and any sources have not been used other than those specified. All the consulted published or unpublished work of others have been clearly cited. I additionally declare that the work and master's thesis have not been submitted for any other previous degree examinations.

München, October 6, 2022

Place, Date

Signature

Kurzfassung

Twitter Daten waren für verschiedene Aufgaben wie Themenklassifizierung und Stimmungsanalyse nützlich. Die Rohdaten von Twitter sind jedoch mehrsprachig und weisen Rauschen und Kurztext auf. Um nützliche Informationen aus Tweets abzufangen, sollten die hochdimensionalen Rohdaten in nieder-dimensionalen Darstellungen kodiert werden. In den letzten Jahren standen verschiedene Aspekte im Mittelpunkt der Forschung, um die Aussagekraft der Tweet-Darstellungen zu verbessern, wie zum Beispiel die Einbeziehung verschiedener Modalitäten von Daten wie Text, Audio, Bilder und Videos. Es gibt jedoch nicht viele Arbeiten zu mehrsprachigen multimodalen Repräsentationen, da große Datensätze von hoher Qualität fehlen. In dieser Arbeit schlage ich ein neues Modell vor, um mehrsprachige und multimodale Darstellungen einzelner Tweets zu lernen, indem ich eine Aufgabe zur Themenklassifizierung einsetze. Mein Modell verwendet nicht nur traditionelle Modalitäten wie Sprache und Vision, sondern profitiert auch von Twitter-spezifischen Modalitäten wie Antworten und Zitaten. Meine These zeigt zunächst, dass das Hinzufügen von Text- und Bildfunktionen für Antworten und Zitate zu den Tweet-Funktionen die Genauigkeit des Modells um mindestens 3% verbessert. Meine Arbeit zeigt auch, dass die Kombination von multimodalen und mehrsprachigen Merkmalen durch die Verwendung von Aufmerksamkeitsmechanismen die Aussagekraft der Darstellung von Tweet-bezogenen Eingaben (Tweet, seine Antworten und Zitate) erhöht, indem die semantische Nähe der Eingabedaten erhalten und die Genauigkeit um zusätzliche 5% erhöht wird. Die Ergebnisse zeigen, wie unterschiedliche multimodale und mehrsprachige Einbettungen innerhalb jedes Dokuments aufeinander abgestimmt erlernt werden können, um ähnliche Klassen einander näherzubringen. Ich erwarte, dass meine Masterarbeit ein weiterer Schritt für weitere Arbeiten ist, die Cross-Modals für Twitter-Daten in mehreren Sprachen kombinieren wollen, um eine Tweet-Darstellung zu verbessern.

Abstract

Twitter data has been useful in different tasks such as topic classification and sentiment analysis. The Twitter's raw data, however, is multilingual and presents noise and is short-text. In order to catch useful information from tweet, the high-dimensional raw data should be encoded in lower-dimensional representations. In recent years, different aspects were the center of research to enhance the meaningfulness of the tweet representations like including different modalities of data such as: text, audio, images, and videos. However, there are not many works about multilingual multimodal representations due to lack of large high-quality datasets. In this thesis, I propose a new model to learn multilingual and multimodal representations of individual tweets by leveraging a topic classification task. My model not only uses traditional modalities, like language and vision, but also profits from Twitter-specific modalities such as replies and quotes. My thesis first shows that adding replies and quotes texts and images features to the tweet features improves the model's accuracy by at least 3%. My work also shows that combining multimodal and multilingual features by using attention mechanisms increases the meaningfulness of tweet's related input (tweet, its replies and quotes) representation by preserving the semantic closeness of the input data and increasing the accuracy with additional 5%. The results demonstrate how different multimodal and multilingual embeddings within each document can be learned in an aligned manner in order to push similar classes close to each other. I anticipate my thesis to be another step for more works that want to combine cross modals for Twitter data in several languages in order to enhance a tweet representation.

Acknowledgement

Firstly, I want to thank my parents, who were a great support for me not only in this project but throughout my life. My brother and my sister thanks as well for motivating me and for being there for me.

Special thanks to Dr. Erick Elejalde for accepting me in L3S and for trusting me working on this project.

Thank you Sergej Wildemann for for providing guidance and feedback in this project and for being so supportive and helpful.

Thanks to all my friends for always accompanying me and encouraging me to give the best of me.

Contents

Chapter 1

Introduction

1.1 Motivation and Problem Definition

Social media play an integral role in most people's daily lives. It has been a great source of information, awareness and building social thought. Nowadays, one can acquire knowledge about, for example, new trends and technologies [GSZ+16]. Also, social media provides facts and data in a consumable way by creating platforms for sharing stories, narratives, and photos. Furthermore, it helps people interact with each others by contacting friends, family, and others even though they live far away. Responding to the huge demand, platforms like Facebook, Instagram, and Twitter are always working on improving their services in order to ameliorate the user experience. The remarkable progress in the technology industry in the last years has further increased the number of users to 4.48 Billions in 2021 [1].

Twitter in particular, which is one of the most active social media platforms, is gaining more users every day. The Figure 1.1 illustrates the evolution of number of Twitter users from 2010 to 2019 as the user base grew from 20 to around 260 millions. Twitter represents an assembly of 500 million tweets posted per day [TCHB12] and 320 million active users in 2021 and is expected to keep increasing up to over 340 million users by 2024 [2].

Internet users on social media spent around 144 min per day in 2019 [Bho20] sharing

[1] https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/

[2] https://www.statista.com/statistics/303681/twitter-users-worldwide/

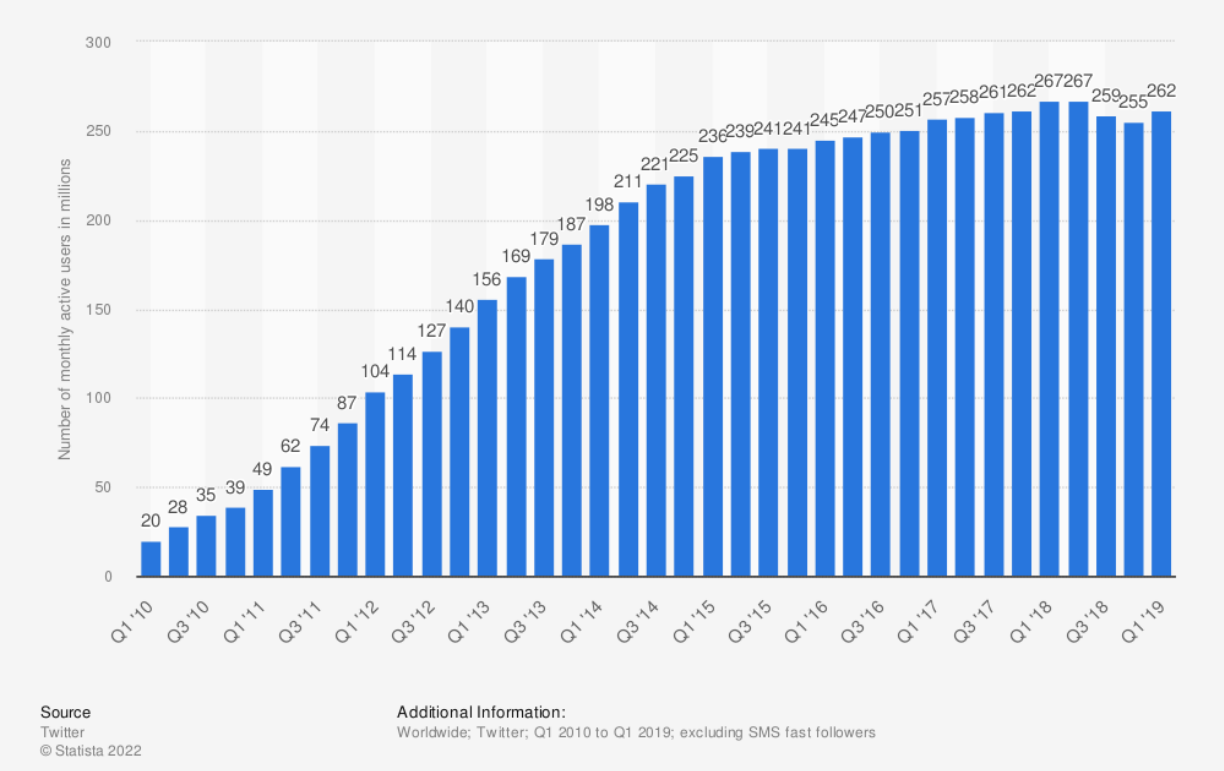


Figure 1.1: Number of monthly active Twitter users worldwide from 1st quarter 2010 to 1st quarter 2019 [3].

images, articles, and videos which led to the generation of a big amount of data. This shared data reflects the user behavior as well as the environment in which they operate and intercommunicate. For Twitter users in particular, posted tweets, comments on tweets and retweets can provide a good idea about people's thoughts, beliefs, political and economical statuses or languages. One can for example understand the public opinion about a certain political party and even predict the results of elections [DPH15]. Twitter presents major advantages in comparison to other social media platforms. You can connect with any person registered on the platform without any exclusivity, whether it is your friend or a famous person like Elon Musk [Fai20]. Twitter is also a place where news spread fast. This leads to a major point: Twitter gives access to a massive amount of data where analysts can outline patterns, popularity of topics and social influence propagation [YW10, LLS17].

Researchers took note of this invaluable information and knowledge source. Twitter data

[3] `https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/`

has been used for different tasks such as topic classification [ISBX17] and sentiment analysis [CHMBE21] have been studied in order to provide a better understanding of patterns and interconnections between all elements of provided data. The raw data, however, presents noise and irregularities. It is also high-dimensional and the relation to the output is complicated to express [KHR+18]. Abstract representations of raw data help in making the expression of the relation between the raw data and an output possible. Selecting the right features from raw data is difficult for many tasks. Therefore, "Representation learning" is becoming increasingly relevant. The aim of this approach is to learn useful information from raw data by leveraging machine learning techniques [LLS20] [Ang20].

In social media, however, more specifically with Twitter, representation learning has some challenges that need to be addressed. In comparison to standard text datasets like TREC-8 [V+99] and IMDb [MDP+11], Twitter's text data contains more noise: Tweets can incorporate spelling mistakes, informal acronyms, sentences can have missing parts such as a verb or subject which can lead to a miss-understanding of the whole text. Tweets are also short texts, as they are restricted to 280 characters.

In recent years, more aspects were the center of research to enhance the meaningfulness of the tweet representations like including different modalities of data such as: text, audio, images, and videos [LLXH22] [LSG+21] [CDFT]. To this end, some solutions were proposed to extract features from multilingual text and vision inputs [CCDMC+20] [BEACC22] [TL19]. These approaches leverage models that were trained on a big number of tweets and/or images to achieve state-of-the-art performance for a wide range of language-vision tasks, such as question answering, image-sentence retrieval, image captioning, visual question answering, sentiment analysis and language inference [MBP+20].

However, these works have several limitations. They are specific to English and do not perform well in other less-resourced languages. This is due to the fact that the state-of-the-art vision-language models are trained on large multimodal datasets that have only English text data, such as MS-COCO [LMB+14] and Flickr30k [YLHH14]. For other languages, there is a lack of large datasets that could be used to pretrain big vision-language models. Also, since collecting non-English multimodal data is costly in terms of time and resources, e.g., due to the lack of the number of native speakers or experts, multilingual vision-language datasets such as Multi30K [EFSS16] and VATEX [WWC+19] are typically small.

Another limitation in the existing multimodal approaches is that they concentrate mostly on two or three modalities, such as combining one sentence and image, but rarely combining several texts and images. Due to these limitations, there is a need to improve Twitter data representation by leveraging multilingual and cross-modal features.

The problem addressed in this thesis is learning representations of Twitter's data. This representation should not only preserve the semantic closeness of the input data, but also lead to a strong predictive capacity for the particular task. This consists of many steps including data gathering, data pre-processing, features extraction and the conception of models in order to perform a specific task, e.g., topic classification.

1.2 Contributions and Thesis Outline

In this thesis, I propose a new model to learn multilingual and multimodal representations of individual tweets by leveraging a topic classification task. My model uses traditional modalities, such as language and vision, but also benefits from Twitter-specific modalities like replies, and quotes. I hypothesize that these additional dimensions distinctive to social media can provide further semantic context to the tweets' representation.

The proposed architecture uses pretrained multilingual transformers to extract language features and pretrained image encoders to extract image features. Deep neural networks are further used to learn the representation of tweets and classify these representations into topics. To test my model, I also propose a multilingual and multimodal dataset "MCMTRA" which spans more than three languages and contains tweet texts, replies, quotes and images as features.

This work is, to the best of my knowledge, the first Deep Learning-based method that leverage both multilingual and cross modalities (replies and quotes to a tweet) to enhance the representation of tweets. The rest of the thesis is structured as follows:

In *Chapter 2*, the required theoretical background is presented so that all the steps are better understood. In *Chapter 3*, methods that formulated the problem, used similar methodologies as my work to a similar problem and works inspired by are presented. In *Chapter 4*, I also describe the process of my dataset collection. I also compare some explored datasets similar to my dataset. In *Chapter 5*, I describe the proposed model. In *Chapter 6*,

experimentation settings and results of the various tests will be presented. In *Chapter 7*, concludes the work and some future directions are presented.

Chapter 2

Background

In this chapter, I introduce some background concepts and methods that will help to later explain the proposed model of Multilingual and Cross-Modal Embedded Representation for Tweets Aggregation. I start with explaining the terminology and defining the notations used in this thesis.

2.1 Traditional Methods

Before I investigate deep architectures which are widely used today for machine learning, it is very important to review the traditional feature learning algorithms first as they help to understand the feature extraction process.

The traditional approach to solve natural language processing(NLP) tasks involves a collection of distinct subtasks. The text corpora need to be preprocessed, focusing on reducing the vocabulary and noise. Punctuation marks and stop words, for example, can distract the algorithm from capturing the main linguistic features required for the task. The next focus is feature engineering, which traditionally involves human understanding of a language. For example, when working on a sentiment classification task, a sentence can be represented with a tree and each node/subtree can be assigned positive, negative, or neutral labels to in order to classify the sentence. Furthermore, the feature engineering part can use other resources such as WordNet [Fel10] to improve resulting features. In the following, I will provide a brief overview of the most basic methods such as N-Grams

[Mar21], term frequency-inverse document frequency (TF-IDF) [1] and one hot encoding.

In N-Grams, the text is broken down into smaller components i.e., tokens consisting of n letters (or words) and each N-Gram w is assigned a probability of occurrence $P(w \mid h)$. This is the number of times w is present in the corpus over the total number of N-Grams h. To estimate the probabilities of N-Grams, maximum likelihood estimation (MLE) [Sti07] is used. For example, to compute a bi-gram probability of a word w_n given a previous word w_{n-1}, the count of words $C(w_{n-1})$ should be computed and then normalized by the sum of all bi-grams that share the same first word w_{n-1}. This can be simplified as shown in the equation 2.1 because the sum of all bi-gram counts that start with a given word w_{n-1} must be equal to the uni-gram for that word w_{n-1}.

$$P(w_n \mid w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})} \tag{2.1}$$

In TF-IDF, the text is converted into a collection of tokens. In this bag of words (tokens), the frequency of occurrence is determined. Frequent words such as articles or linking terms have a high recurrence in documents, corresponding to a high term frequency (TF). A simple approach is to remove these so-called stop words. Inverse document frequency (IDF) is then used to determine the weight of words in all the documents of the corpus. All this can be summarized in the equation 2.2.

$$tfidf(t,d) = tf(t,d) \times log(\frac{N}{(df+1)}) \tag{2.2}$$

With N number of documents, $tf(t,d)$ being the frequency of the term t in a document d and df being the document frequency. During the query time, if a word which does not occur in the vocabulary, the df will be 0. As the division by zero is not defined, the value is smoothed by adding one to the denominator.

Finally, in one-hot Encoding, each word is represented in a binary way. A vector will be filled with zeros and ones depending on the location of a word in a given text. At the end, a matrix of $n \times m$ is outputted, where n is the number of unique words and m is the number of words in the given text.

[1] `https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/`

Table 2.1: Representation of the transition from labeled data to a one-hot encoding representation[Per21].

Food Name	Categorical	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

Table 2.2: Label Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

Table 2.3: One hot encoding

The Table 2.1 shows how the categorical representation of data can be changed to binary. This allows to represent each word in the vocabulary ("Apple", "Chicken", "Broccoli") with a unique one hot vector representation as follows: Apple: $\begin{bmatrix}1 & 0 & 0\end{bmatrix}$, Chicken: $\begin{bmatrix}0 & 1 & 0\end{bmatrix}$ and Broccoli: $\begin{bmatrix}0 & 0 & 1\end{bmatrix}$. This method is easy to implement as it can represent every word value in a single vector.

These traditional approaches have disadvantages [SVS+14]. External resources which often consist of manually created information stored in large databases are needed for complex tasks such as a rule-based machine translation in order to perform well. Also, preprocessing steps in traditional NLP force the elimination of useful information such as punctuation or tense form in the text in order to make the learning feasible by reducing the vocabulary. That is why more sophisticated methods have been introduced.

2.2 Encoder and Decoder

The encoder decoder model is used in a variety of tasks such as image captioning [XWD+19], sentiment analysis [PMLP18], machine translation [SVL14] and video captioning [VRD+15].

This model is composed of three main components: encoder, decoder and hidden state as shown in Figure 2.1 The encoder decoder model have an input sequence where it is read in entirety and encoded into a fixed-length internal representation (hidden state). Then a decoder network outputs words using this internal representation until the end of the sequence token is reached.

The encoder is composed of a stack of several units, where each one accepts a single element

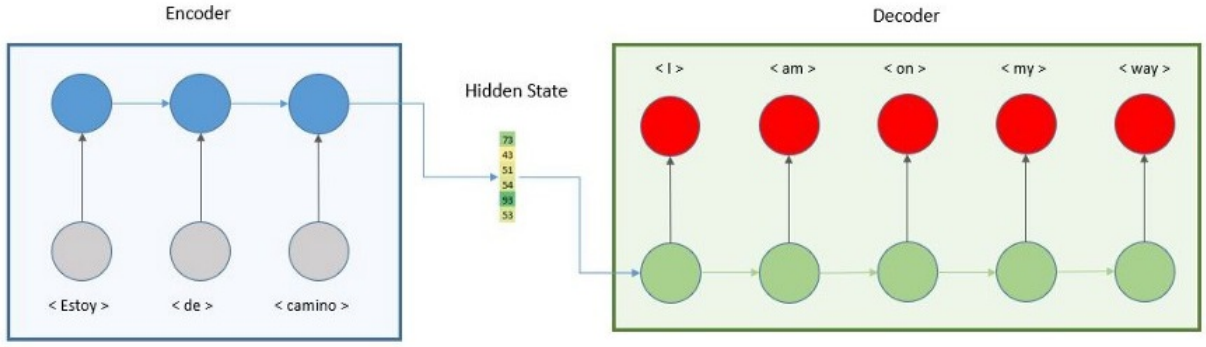


Figure 2.1: Encoder Decoder Model [Kos22].

of the input sequence and propagates it forward. In a question answering problem, for example, the input sequence is a collection of all words from the question. Each word is represented as x_i where i is the order of that word. The hidden states h_i are then computed using the formula 2.3:

$$h_t = f(W^{(hh)}h_{t-1} + W^{(hx)}x_t) \tag{2.3}$$

where $W^{(hh)}$ and $W^{(hx)}$ are the weights applied to the hidden state and the input vector respectively. The hidden state vector is an encapsulated version of the information for all input elements in order to help the decoder make accurate predictions.

The decoder is a stack of several units, where each one predicts an output y_t at a time step t. In the question-answering problem, the output sequence is a collection of all words from the answer. Each word is represented as y_i where i is the order of that word. Any given output y_i at time step t is decoded using the formula 2.4:

$$y_t = Softmax(W^s h_t) \tag{2.4}$$

where the hidden state at the current time step together with the respective weight W^s. Softmax is used to create a probability vector which will help determine the final output (e.g., word in the question-answering problem). Understanding encoder-decoder models is essential for advances in NLP because it is the seed of attention models and transformers. In the next section, I will continue exploring the attention mechanisms by leveraging an encoder decoder structure.

2.3 Attention Mechanisms

The idea behind the attention mechanism is to enable the decoder to utilize the most relevant parts of the input sequence flexibly, and thus providing a weighted combination of all the encoded input vectors. In this case, the highest weights will be attributed to the most relevant vectors.

The attention mechanism was firstly introduced by Bahdanau et al. [BCB14] to solve the problem of bottleneck when the length of the encoded vector is fixed, thus the decoder would have limited access to the information provided by the input. This will be more problematic when the sequences are complex because the representation dimensionality is forced to be the same as for simpler sequences. The attention mechanism in this context considers 3 steps of computations. First, the alignment scores are calculated using the Formula 2.5:

$$e_{t,i} = a(S_{t-1}, h_i) \tag{2.5}$$

where h_i is the encoded hidden states and S_{t-i} the previous decoder output. The alignment model is represented by the function $a(.)$ and $e_{t,i}$ is the computed score. After that, the weights are calculated using a Softmax function.

$$\alpha_{t,i} = Softmax(e_{t,i}) \tag{2.6}$$

Finally, the context vector is outputted at each time step using a weighted sum of all, T, encoder hidden states.

$$c_t = \sum_{i=1}^{T} \alpha_{t,i} h_i \tag{2.7}$$

In this case, the model projects first s and h into a common space, then applies a similarity measure (e.g., dot product) as the attention score. The score computation problem can then be formulated as follows in the equation 2.8:

$$e_{t,i} = f(s_i)g(h_i)^T \tag{2.8}$$

In this case, $g(h_i)$ has to be computed $\mathbf{m}$ times and $f(s_i)$ have to be computed n times to get the projection vectors and then the score $e_{t,i}$ can be computed using matrix multiplication. This is essentially the approach proposed by [VSP+17], where the two projection vectors are called query (for decoder) and key (for encoder), which is well aligned with the concepts in retrieval systems.

The generalization of the attention mechanism came with the use of three main components, namely the queries (Q) the keys (K) and the values (V). Comparing these three components to the attention mechanism as proposed by Bahdanau et al.[BCB14], then the query would be analogous to the previous decoder output, S_{t-i}, while the values would be analogous to the encoded inputs, h_i.

Taking the example where the vector query $q = S_{t-1}$ is matched against a database of keys to compute a score value. This can be computed as the dot product of the specific query under consideration with each key vector k_i:

$$e_{q,k_i} = q\dot{k}_i \tag{2.9}$$

Then the weights are calculated in the following equation:

$$\alpha_{q,k_i} = Softmax(e_{q,k_i}) \tag{2.10}$$

If I combine all the equations cited above, the attention function can be calculated as a weighted sum of the value vectors v_{k_i}. Each value vector is paired with a corresponding key. The attention function problem can then be formulated as follows:

$$attention(q, k, v) = \sum_i \alpha_{q,k_i} v_{k_i} \tag{2.11}$$

Self-attention is a particular case of attention systems. In this case, Q, K, V are usually from the same source. This was very helpful for cases like unsupervised language model training such as GPT [RNS+18]. To explain self-attention, the simple example shown in Figure 2.2 is considered. Here, three inputs are provided with a dimension of four. The searched representation should have a dimension of three, so the set of the weights must have a shape of 4×3. In a neural network setting, these weights are usually small numbers, initialized randomly using an appropriate random distribution like Gaussian distribution. To derive the key of the first input, the initialized weights are used as follows:

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 4 & 4 & 0 \\ 2 & 3 & 1 \end{bmatrix}$$

[2] https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

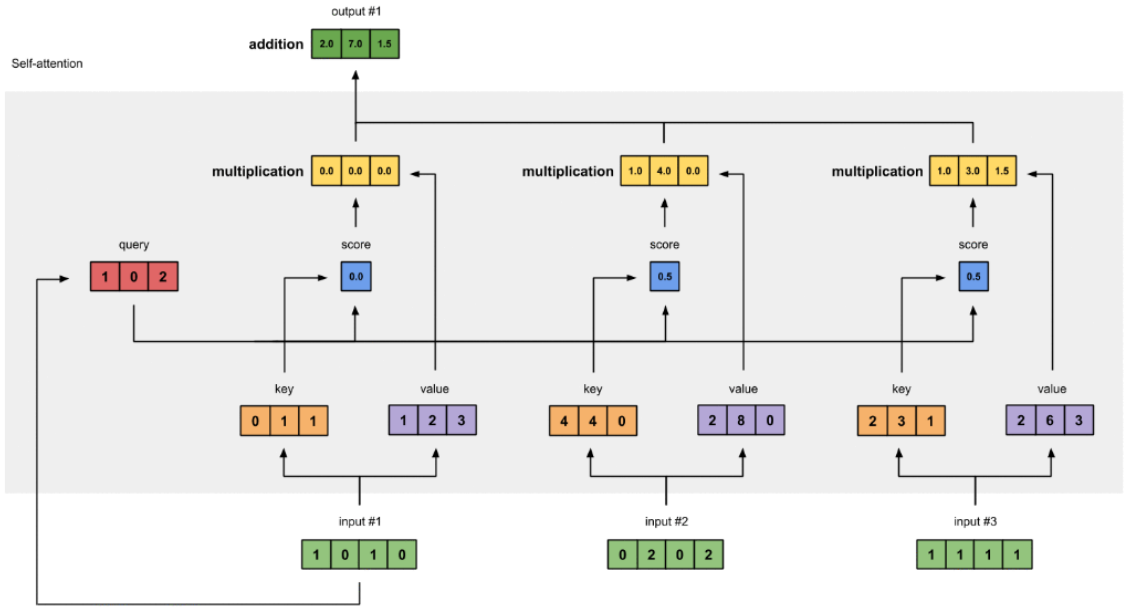


Figure 2.2: illustration of different steps of self-attention computation [2].

The same procedure is held to compute the values and queries using their corresponding initialized weights. The next step will be to calculate the attention scores by taking the dot product between the query and all keys, including itself. In this case, the three attention scores are:

$$\begin{bmatrix} 1 & 0 & 2 \end{bmatrix} \times \begin{bmatrix} 0 & 4 & 2 \\ 1 & 4 & 3 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 4 & 4 \end{bmatrix}$$

Once the scores are computed, passing them through a Softmax function will give:

$$Softmax(\begin{bmatrix} 2 & 4 & 4 \end{bmatrix}) = \begin{bmatrix} 0.0 & 0.5 & 0.5 \end{bmatrix}$$

The next step would be to multiply the Softmaxed attention scores with values in order to get weighted values. The element-wise sum of these weighted values creates the output. The found result vector [2.0, 7.0, 1.5] is for output 1 which is based on the query representation from Input 1 interacting with all other keys, including itself. So the same computation steps should be repeated for input 2 and 3.

2.4 Image Representation

Image representation is based on two paradigms: Residual representations and Shortcut Connections.

A residual block is a stack of layers set in such a way that the output of a layer is taken and added to another layer deeper in the block. To get a better representation for an image in an image recognition task, VLAD [JPD+11], for example, encodes the residual vectors with respect to a dictionary, and Fisher Vector [PD07]. Another approach can be based on preconditioning method [Sze90] which works with variables that represent residual vectors between two scales.

Shortcut connections [B+95] [Rip07] as the name suggests, link between two distant layers without involving the set of layers between them. An example of performing shortcut connection is in "highway networks" [SGS15] where gating functions are used [HS97]. These gated shortcut learns residual functions and are never closed so that information is always passed through with additional residual functions to be learned.

2.5 Pre-trained Models

Extracting the most significant features for the proposed model relies on a very important component, which is the use of pre-trained models that achieved state-of-the-art results on various vision and language tasks. These pre-trained models are easy to incorporate into other works since their architecture is available to use and requires less training because their weights are already optimized.

2.5.1 Vision

To fulfill the task of extracting vision features, I use ResNet-50 [HZRS16] which is based on residual connections between layers. In plain deep networks, by simply stacking more and more layers, gradients get small, leading to the problem known as vanishing gradients. ResNets came to solve this problem with connections that skip over layers without presenting any further parameters. Hence, residual links helped in reducing loss, preserving knowledge gain and boosting performance during the training phase. Here, skip connections are used so that the output of a layer is a convolution of its input plus its input. A block diagram of

the ResNet model's architecture is shown in Figure 2.3.

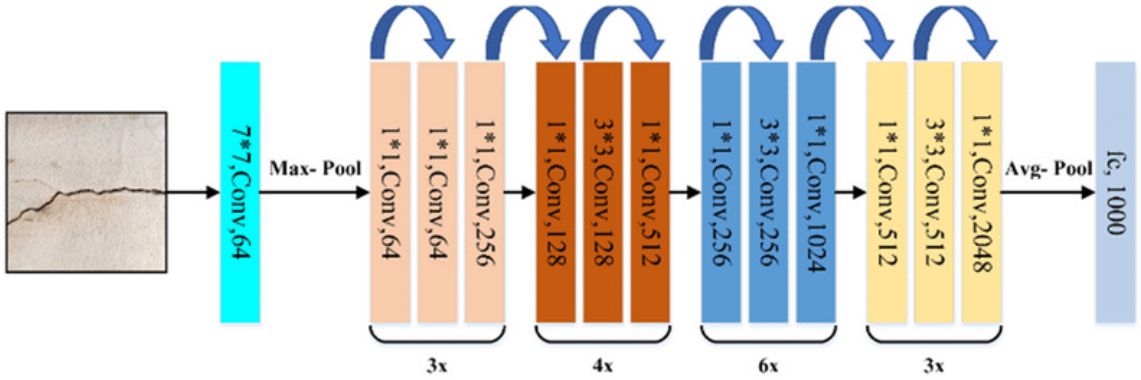


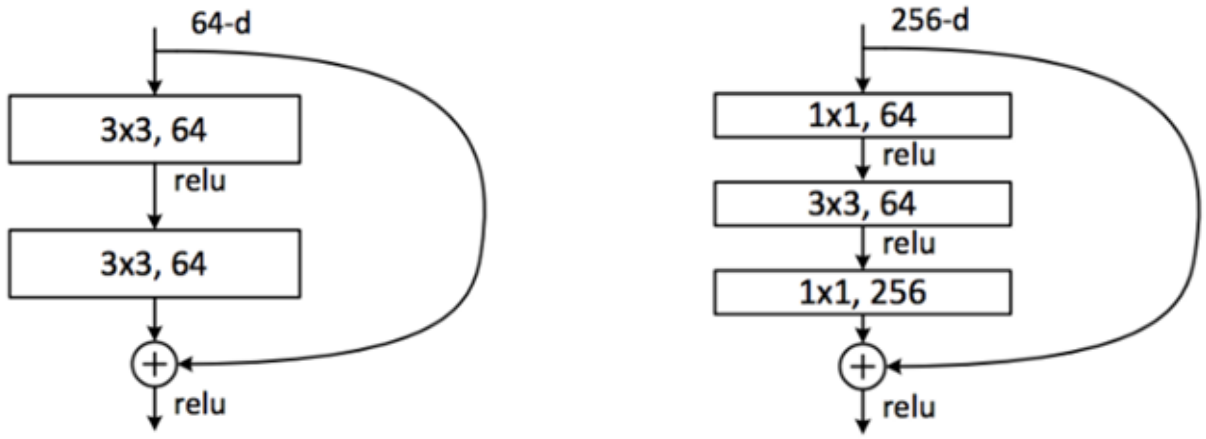
Figure 2.3: Illustration of the ResNet-50 architecture [AAJ+21]

ResNet-50 operate with 7×7 convolution layer as input with a max pooling layer after it and fully-connected layer of size 1000 as output with an average pooling layer before it. The rest of the 48 layers are 16 sets of a) a 1×1 convolution, b) a 3×3 convolution and c) a 1×1 convolution. The first three sets have as depth (this is the depth of the first two convolutions of each set while the third one has 4 times that depth) 64, the next four 128, the next six 256 and the final three 512. The input of each set is transformed to the next input using shortcut connections. Then the two inputs are concatenated using the residual function:

$$y = F(x) + x \tag{2.12}$$

where y is the input of the next set, $F(x)$ is the output of the set and x is the input of the set.

He et al.[HZRS16] introduced two building blocks with which they constructed their different versions of ResNets: basic block for ResNet34 and bottleneck block for ResNet50, 101 and 152. Bottlenecks are used in deep ResNets to reduce computation complexity. They reduce the channels of the input before performing the expensive 3×3 convolution, then project it back into the original shape. Both blocks are illustrated in Figure 2.4



(a) basic building block

(b) bottleneck building block

Figure 2.4: Illustration of residual building blocks: A basic building block for ResNet-34 and a "bottleneck" building block for ResNet-50/101/152. [HZRS16].

2.5.2 Language

Recent pre-trained language models, such as ELMo [PC+19], BERT [DCLT18], GPT2 [RWC+19], XLNet [YDY+19], RoBERTa [LOG+19] and ALBERT [LCG+19] have two keys to their success: effective pre-training tasks over a large language corpus, and the use of Transformer for learning contextualized text representations.

I chose to use RoBERTa which is an improved version of Bert [DCLT18]. RoBERTa is one of the best transformers as it is trained longer, with bigger batches, on longer sequences in comparison to Bert. As an example, I refer to the Benchmark study for online fake news detection [KKA+21] of several pre-trained models where RoBERTa achieves notably better performance than ELMo, Bert and DistilBERT [SDCW19].

I use in particular three versions of XLM-Roberta [CKG+19]: (i) xlm-roberta-base (ii) Twitter-xlm-roberta-base and (iii) Twitter-xlm-roberta-base-sentiment [BEACC22]. This model with its different variants represent the multilingual version of RoBERTa trained on more than 100 languages.

To ensure a good training, XLM-Roberta leverages the use of transformer model [VSP+17] trained with the multilingual Masked Language Models (MLM) objective [DCLT18]. From each language, streams of text are sampled, and then the model is trained in order to

predict the tokens in the input that were masked. Sub-word tokenization is applied on the raw text. After sampling batches from different languages a large vocabulary is used to train the model which composed of 12 layers i.e., Transformer blocks and where the hidden size is 768, the number of self attention heads is 12 and the number of parameters is 270M.

To guarantee the scale to a hundred languages, XLM-Roberta leverages the model from [LC19] trained on Wikipedia text in 100 languages and along with CommonCrawl dumps for English and twelve other languages. This has helped increase performance, especially for low resource languages.

Even though the versions (ii) and (iii) of XLM-Roberta use the same architecture as (i) have another additional particularity which is trained on $\approx$ 198M multilingual tweets and fine-tuned for sentiment analysis on 8 languages. The Figure 2.5 shows the most frequently used 30 languages.

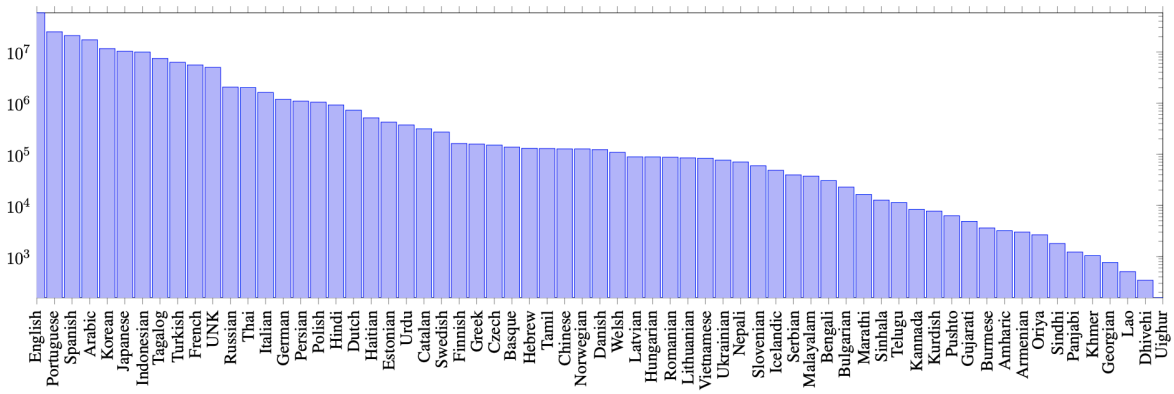


Figure 2.5: Distribution of languages of the 198M tweets used to fine-tune the Twitter-based language model [BEACC22].

It is to notice that English, French and German languages are located within the 15 first commonly used languages. This means that these languages are considered high resource and the used pre-trained model will have no issue extracting features from the given text tweets. Also, the fine-tuning for Twitter and sentiment analysis task takes in consideration these three languages. This fine-tuned version of the model was proved to be more consistent for specific Twitter tasks than its general version [BEACC22].

Chapter 3

Related work

In the last years, many methods for data representation have been explored. In addition, many types of datasets have been created and used in Natural Language Processing (NLP) tasks. In this chapter, an overview of the methods used for data representation as well as the most popular social media datasets will be presented.

Data representations techniques can be classified according to different criteria, such as the type of triggered features or the adopted method type. Moving forward, the methods that were the subject of my research will be presented based on their approaches and multilingual support.

3.1 Approaches to Data Representation Learning

Multiple approaches of learning data representation were considered in the literature. Some of them adopted an unsupervised manner for resolving tasks like Masked Language Modelling [CLY+20] and Masked Region Modelling [LLXH22]. Unsupervised data representation learning in this case is based on the concept of fill in the blanks of a problem statement. A masking of certain percentage of the input which can be typically either language or visual will be held. The model is then expected to predict masked words or regions in an image based on other words in a sentence or other regions in a visual data. Because the representation of the masked word is learned based on the words that occur on the left as well as right side, such a training scheme makes the model of this approach bidirectional in

nature. This will help understands how transformers like Bert [DCLT18] and RoBERTa [LOG+19] are trained. In fact, these Bidirectional Encoders understand the meaning of ambiguous language in text by using surrounding text to establish the context, as shown in Figure 3.1.

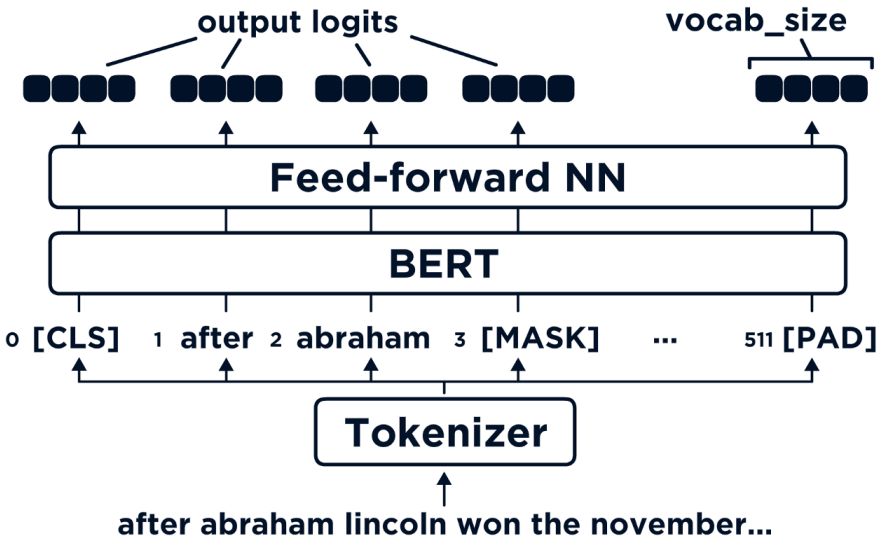


Figure 3.1: Masked-Language Modeling With BERT[1].

Another approach is based on supervised learning for resolving tasks like Sentiment and Emotion Detection [CHMBE21] [CCW19], text classification [YS19] and topic classification [KLH22]. In the studies cited above, all models used Deep Learning-based methods to create different representations of documents into classes.

Furthermore, they often leverage multi-modalities, to enhance meaningfulness of the extracted features. [CHMBE21] proposed a solution that extracts different visual features such as objects, facial expressions and places along with contextual text representations to predict the sentiment in a tweet. Few of these methods looked at more than two modalities. Multiple approaches showed the benefits of combining them in order to get a better representation of the data. [CCW19] developed a fusion mechanism by adding a modality fusion layer that performs weighted average to the vectors and feeds them to a classification layer to yield the final result. This helped further improve the final representation of a tweet by refining the representation of each modality, which proved to be significantly more effective than simply concatenating three types of features i.e., text, image, and image attribute

[1] https://towardsdatascience.com/masked-language-modelling-with-bert-7d49793e5d2c

features. [KLH22] takes into consideration more features such as hyperlinks, author features and entities to classify tweets into topics.

3.2 Multilingual and Multimodal Representation Learning

3.2.1 Multimodal Representation Learning

Learning a multimodal data representation has attracted a lot of research in the last years and have added a lot of value to many vision-language tasks such as cross-modal retrieval [KSZ14, FFKF17, WYW+, VKFU15, NHK17, LCH+18, HCHH20], image and video captioning [RMM+17, KTS+14, XMYR16, DL17, WWC+19], visual question answering [AAL+15] [GKSS+17] [AHB+18] and multimodal semantic indexing and information extraction for multimodal content [HYL+17] [HCC+19]. In the references cited above, the most common practice is to learn a visual-semantic embedding (VSE) space[WYW+16]. The contents in these different modalities such as images, text queries and videos are projected in the same embedding space in order to interact with each other. In this case, the items of different modalities can be directly compared to each other in order to measure the content similarity. If multimodal contents are correlated semantically, they are mapped in proximity.

Although much research attention was attracted to vision-language tasks, no significant development was conducted to improve non-English data representation. Most of the multimodal datasets such as MS-COCO [LMB+14], Flick30K [Fer14] and Visual Genome [KZG+17] are annotated in English, but only 7% of the world population are native English speakers. It will be difficult yet costly to apply and generalize the current vision-language models to the non-English task problems.

3.2.2 Multimodal Multilingual Representation Learning

Some recent works have investigated multilingual vision-language models [EFSS16] [WWC+19] which resulted in a small dataset of 30,000 images and captions in 4 languages for Multi30K [EFSS16] and 20,000 videos and captions in two languages in VATEX [WWC+19] [HKL+19]. Furthermore, other researches were held to learn k-view representations. A relevant work in [GSKL17] which extended the work of [CLC17] used images as

pivot between two languages by learning a common representation for images and their descriptions in German and English and then evaluated the model with image-description ranking for both language and semantic similarity with the pair image and descriptions in English. The common approach in prior work in order to learn a representation for multimodal and multilingual data is the projection of image representation to the textual representation of aggregated data in the same embedding space for alignment.

The references cited above let me conclude that a large scale corpora is important to learn a robust multilingual multimodal representation. In other words, the high cost of preparing visual data along with multilingual annotations leads to sparse and size limited multilingual multimodal data. The challenge is then to develop annotation-efficient methods that learn from weakly-labeled.

3.3 Challenges in Data Representation for Twitter Classification

Trying to solve a topic classification problem in particular for Twitter data requires dealing with many challenges. One of many challenges is labelling the data. The human annotations of a random sampled set of tweets will be a very inefficient way to approach this task due to the need of large cognitive load of annotations. [YKSG14] proposed a solution by sampling tweets taking into consideration topic priors to get a first weakly relevance to a specific topic. This operation was held by training a logistic regression-based model to derive hashed N-Grams based features from the tweet text. The human annotations can then confirm the results. Many other researches [GD17], [GEM16] concentrated their analysis on the structure of the tweet text. This investigation aims to detect what are the most important parts within a tweet that can point out to its overall meaning, such as hashtags and named entities. [GD17] investigated the effect of hashtags segmentation and harmonization on the process of clustering. [GEM16] went a step further by examining domains in social semantic web [2] and proposing a framework that links tweets coming from different domains and languages. [KJTS19] developed Zero shot hashtag recommendation system to overcome the problem of the unfeasibility to collect data for all possible hashtags labels. This proposed paradigm learns the relationship between the embedding space of

[2] https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers

labels and semantic space of tweets. Since my work responds to the data representation problem by solving a topic classification task, I was inspired by the works cited above to use hashtags as a solution to annotate my dataset.

3.4 Attention Mechanism in Representation Learning

Combining different features in their multimodal forms can lead to a better learning of the data representation [YYS+21]. Also, the way of combining different modalities in a way that reflects the overall meaning is a very significant factor [DLA17, ZLY21, ZZS+19, WWA+19, TL19].

Existing methods for multimodal classification tasks can be categorized according to the fusion stage, i.e., early data fusion, intermediate representation fusion, and late decision fusion [ZLY21].

The purpose of early data fusion is to integrate information from multiple data sources or views into one feature vector. It investigates each modality separately without considering interactive relationship among modalities, which neglects the complementary information among the modalities. The main disadvantage in this approach is that the relationship between modalities cannot be fully captured because of the redundant noises. In this context, [PCCH16] first used convolutional neural networks to learn the representations of images and texts, then used kernel learning classifiers to combine multi-view information.

The intermediate representation fusion works on extracting individual characteristics from each modality in order to fuse them into a joint representation. The main concern here is to mine common inter-modal information in order to achieve a higher accuracy with the complementary effect between the modalities. In the literature, intermediate representation fusion is the most used method in multimodal classification tasks in order to capture the relationship between the modalities for learning more discriminative representations. Two main approaches were considered: direct concatenation[HF18] and use of attention mechanism to generate joint representations. Most researches presume that there is a one-to-one correspondence within the text-image pair [YLJY16]. [YLJY15] explained how different modalities are consistent for expressing the same sentiment, and a consistency constraint was added in order to enforce the similarity between prediction functions of each modality. [YCJL16] proposed a tree-structured model to explicitly align textual words and

visual regions for learning joint representations.

Late decision fusion, as its name indicates, is performed at the decision-making stage. First, the model predictions are treated independently, then final results takes the outputs from each model to get single modal information. The late decision fusion has better robustness and generalization capacities because errors from multiple models are dealt with independently and thus by capturing the unique intra-modal information. Several works were conducted in this direction, such as [VWZL19] who proposed a neural network-based model to learn the weight coefficients after concatenating different modalities. In my work, I leverage the findings of this section to learn intra-modal and inter-modal features.

Chapter 4

Dataset

In this chapter, I introduce the project's multilingual cross-modal dataset. I will describe in depth the specifications of my dataset. Afterwards, I will present in short some existing datasets and make a comparison between those and my proposed dataset.

4.1 MCMTRA Dataset Sources

I collected data from Twitter, in particular posted tweets and corresponding interactions such as retweets, likes, quotes, replies, and profile information from 956 different news outlets starting from the date of 01.01.2022 till 30.07.2022.

I choose news outlets as a data source because they belong to better defined topics and have a great impact on people's reactions [RP19]. Not to mention that, tweets posted by news outlets are better structured than those posted by regular users. More specifically, I chose national and even international news outlets such as "Bild", "Euronews" and "Le Monde" because studies on user engagement with news on Twitter [DGG+21] proved to be larger this way i.e., these accounts show a greater number of replies and quotes compared to regular user accounts. From these news outlets, 362 are located in Germany, 259 are in France and 289 are related to English communities such as the United Kingdom.

Because the task is to have topic specific embedding and tweets are short in nature, more modalities are needed to enrich the document representation. To this end, since news tweets consistently generate interactions such as replies and quotes, they were used as additional

modalities. Then these data instances will be used to learn representations of different documents and analyze relations between them.

4.1.1 Data Collection Process

Collecting data from Twitter can be done either: (i) directly from API. or (ii) using scraping software packages. I used the new version of Twitter API called Twitter API V2 which offers new and more detailed data objects. I used also python packages such as "tweepy [1]", "snscrape [2]" and "twarc v2 [3]" to build a data collector platform. This platform enables the scheduling of queuing jobs for retrieving the needed data and processing them in the background with workers. This part of the work can be extended by adding a user interface to facilitate its usage for non-technical persons. I used REST API, which enables the user to return contents stored on Twitter servers that corresponds to a query request. I am using the essential access level of Twitter. This mode can provide a restricted amount of data in comparison to Academic Research access.

Because of the restrictions imposed by Twitter's policy, I chose to use different endpoints, and packages depending on the maximum allowed number of requests per hour, number of returned results per request. The used configuration and limits are presented in Table 4.1.

Table 4.1: representation of the retrieved data elements and limitations

Element	Used package	Limit
Tweets	Snscrape	Up to 500 results/request with paginator
Replies	Snscrape	Up to 500 results/request with paginator
Quotes	Snscrape	Up to 100 results/request with paginator
Likers	tweepy	75 requests per 15 minutes
Retweets	tweepy	75 requests per 15 minutes
Profiles	tweepy	900 requests per 15-minute
Followers	twarc2	1000 results/request 15 requests per 15-minute

[1] `https://www.tweepy.org/`

[2] `https://github.com/JustAnotherArchivist/snscrape`

[3] `https://twarc-project.readthedocs.io/en/latest/twarc2_en_us/`

I chose SQLite database, which is a database engine that does not require a separate server process and allows the user to access the database using a nonstandard variant of the SQL query language. Although I retrieve the profile of users that retweet, like and follow a certain tweet or profile, I did not use them in the next steps of my work because of the limits imposed by the Twitter API. I chose to use tweets, replies, quotes and number of likes and their corresponding images if they are provided as principal features in this work because these elements can be retrieved faster than the others. Also, I want to concentrate on the data and metadata, and see what they bring to the final representation of tweets and the user aspect will not be taken into consideration.

The remaining data stored in the database is organized in four tables; "profileDescription", "quotesToTweet", "repliesToTweet" and "searchTweets". The proposed structure of the database is presented in the figure 4.1.

The process of getting the data is as follows: For every news outlet username, first the profile names of the news outlets were given as inputs, then profile information was retrieved and stored in the "profileDescription" table. Afterwards, using the username and user ID of each retrieved newspaper's profile, corresponding tweets along with other information relevant to that specific tweet were collected. I store the latter in "searchTweets". "repliesToTweet" and "quotesToTweet" are filled with data relevant to replies and quotes related to each tweet.

4.1.2 MCMTRA Specifications

The initial data I got from the 956 news outlets includes 609,575 tweets, 1,042,456 replies and 584,870 quotes.

In the following, I present the numbers of different elements of the tweet representation based on multiple criteria. Since the hashtags are used for the labelling process, I first filter the initial data and keep only the documents whose tweet text includes at least one hashtag. The next section 4.2 explains the topic assignment by using the hashtags. Afterwards, I define twelve topics, where each topic includes a list of hashtags. Not all available hashtags belong to one of the topics because only the top 1000 most common hashtags were considered. This number was empirically chosen. I also filtered the data based on whether the tweet includes an image. The resulting numbers are visualized in

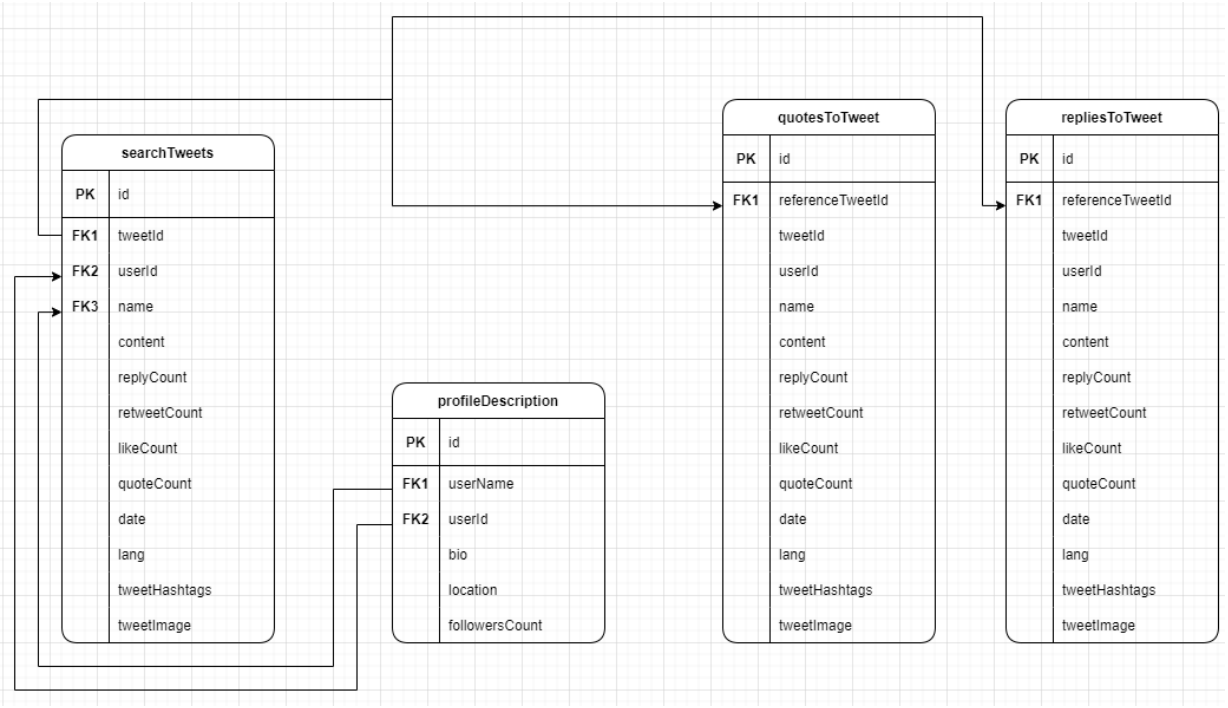


Figure 4.1: MCMTRA Database schema

Table 4.2.

Table 4.2: Numbers of different Elements of Tweets based on hashtags(H), topic(T), and image(I) selection criteria.

Criteria	Tweets	Replies	Quotes
Initial numbers	609,575	1,042,456	584,870
H	181,227	231,596	163,456
H and T	36,049	55,436	32,737
H, I and T	6,693	11,933	7,940

The stated numbers visualized in Table 4.2 show that only around 30% of the tweets actually includes at least one hashtag in the tweet text. Only 20% of those documents with hashtags are assigned to a topic. After applying the image availability criterion on the tweet, only 6,693 tweets, 11,933 replies and 7,940 quotes are left.

I have built two sub-datasets for the test and validation phases of the model. The first one that I am going to note as MCMTRA_1 takes into consideration only the availability

of hashtags in the tweet texts and that every document should be given a topic. In this case, I get 36,049 documents, a number of replies equal to 55,436 and a number of quotes equal to 32,737. The second sub-dataset which I am going to note as MCMTRA_2 takes into consideration the availability of at least two replies per document and one quote per document. In this case, I get 13,056 documents, a number of replies equal to 52,123 and a number of quotes equal to 26,767. I chose these two levels of constraints to explain if the interactions to each tweet can enrich the overall document or not.

My dataset includes tweets mainly from German, French, and English, but also a minority of other languages such as Spanish or Russian. the "Other" row also includes tweets whose language cannot be identified. An overview of the distribution of different languages is found in the Table 4.3. Overall, the percentage of worked with languages i.e., German, English, and French are comparable in size. It is to mention, that the highest number of tweets and replies is in German language. English language has the most quotes. Quotes percentage is relatively high in other languages besides French, English, and German with a percentage of 13.76%.

Table 4.3: Percentage of different languages in MCMTRA based on the language of the tweets, replies, and quotes

Language	Tweets	Replies	Quotes
English	28.94	28.47	38.56
French	28.17	25.05	21.65
German	37.20	42.15	26.03
Other	5.69	4.33	13.76
	100	100	100

4.2 Hashtag Classification

The task here is to annotate the data. Manual annotation methods are costly as they rely on experts and need a lot of time to be executed. That is why the need for an automatic and efficient way to label the data is required. I used the finding in [HAR14] which proved that labels generated by experts matched with hashtag labels in over 87% of Twitter messages, which indicates that hashtags are indeed good labels. Before removing hashtags, I filter

the tweets based on the criteria of whether they contain hashtags. I consider only tweets that have at least one hashtag to get at the end 181,227 tweets. From these tweets, I search for the 1000 (empirically chosen) most common hashtags. It will be very difficult to analyze 1000 classes (one for each hashtag). Also, not enough data samples are available for each hashtag, which will prevent the model's learning process. That's why I manually classify these hashtags into one of 12 classes("war", "AutomotiveIndustry", "corona", "sport", "politics", "fun", "history", "economics", "promotion", "social", "weather" and "technology"). To achieve this task, I consult the first 50 tweets that includes that specific hashtag and decide in which topic it should be included. I exclude from these most common hashtags the ones that could not be mapped such as names of countries, cities("Bremen", "Allemagne") and generic hashtags that does not reflect the idea of the tweet such as "FaitsDivers" or "CeJourLa". The reason behind this is that these hashtags are used in too many topics. For example, if posted tweets from a German sport newspaper always include the hashtag "Germany", the model will learn this as a relation between tweets including this hashtag and the topic.

After this preliminary manual classification of the hashtags into topics and to help further expand the size of the training data by 10%, I implemented a co-occurrence matrix of the hashtags assigned to each topic. This co-occurrence matrix considers all hashtags included in the tweets. Then I apply a 40% rule to get most relevant related hashtags. This value was empirically chosen. This rule considers the hashtags that co-occurred with the hashtag in question when the number of co-occurrences is greater than 40%.

Table 4.4: Example of co-occurence matrix for the hashtag "UkraineInvasion"

Main hashtag	Hashtags cooccurred with the main hashtag				
UkraineInvasion	Ukraine	UkraineRussia	Putin	UkraineKonflikt	Russland
69	43	28	12	8	3

The Table 4.4 shows an element of the co-occurrence matrix where the main hashtag "UkraineInvasion" co-occurred with the hashtags "Ukraine", "UkraineRussia", "Putin", "UkraineKonflikt" and "Russland". Since the main hashtag occurred 69 times among tweets and the hashtags "Ukraine" and "UkraineRussia" co-occurred 43 and 28 times with the main hashtags respectively, applying the 40% rule will give as output these two percentages: $43/69 \times 100 = 62.31\%$ and $28/69 \times 100 = 40.57\%$. These two percentages are greater than

40% so the corresponding hashtags will be added to the topic's vocabulary where the main hashtag was defined.

4.3 Existing Datasets and Comparison

To understand the advantages and disadvantages of the proposed dataset, it is important to compare it to other datasets used for representation learning.

I will categorize datasets in the context of data representations tasks in two categories: multimodal and multilingual multimodal. One of the most famous image-text datasets is **Flickr30K** [YLHH14] which developed a method to use the visual denotations of linguistic expressions to compute the denotational similarities. This was useful in many tasks such as image captioning [GWCC17], object detection [SCN19] and multimodal retrieval [MXY+14]. This dataset is composed of 31,783 images and 158,915 image-text pairs (5 sentences per image). **MVSA-S** and **MVSA-M** [NZPS16] which proposed a set of manually annotated image-text pairs collected from Twitter. This dataset was used for tasks like sentiment analysis [CHMBE21] and contained 4,869 and, 19,598 image-text pairs from Twitter.

For multilingual multimodal datasets **Multi30K** is considered, introduced by [EFSS16] which is a multilingual version of **Flickr30K** supporting German, French, English and Czech languages. Two techniques of collecting and storing the data were used: (i) collect the descriptions of an image and translate this description to another language. (ii) collect descriptions for every image from different resources. This dataset was used for tasks like multilingual captioning [NTN20], image-text matching [WLHL18].

Table 4.5: Dataset's comparison

Dataset	Number of samples	Sentence/Image	Language
Flickr30K	29,000	5	en
MVSA-S	4,869	1	en
MVSA-M	19,598	1	en
Multi30K	29,000	5	en,fr,de,cs
MCMTRA	36,049	≈ 0.25	de,fr,en,other

As it is shown in Table 4.5, **Flickr30K** and **Multi30K** have more image-text pairs in

comparison to the proposed dataset, but the main advantage present in **MCMTRA** is that within a single document several features are provided. In fact, each document is composed of a tweet along with its corresponding replies and quotes. Each of these document parts is also accompanied by its image, if available. This is even more obvious when comparing **MVSA-S** and **MVSA-M** to **MCMTRA** since the latter has almost double the number of samples in comparison to **MVSA-S** and nine times in comparison to **MVSA-M**. Even when adding more constraints on the proposed dataset such as the availability of an image with the tweet text, **MCMTRA** still have a greater number of samples in comparison to **MVSA-S**. Overall, the other datasets concentrate on how to enlarge the number of samples, but they only use two modalities within each document. On the other hand, the proposed dataset responds to the problem of lack of modalities within each document by adding not only images to their corresponding tweets but also interaction to tweets such as replies and quotes related. In chapter 5, I take a deep dive into the organization of these different features into a model to learn different modalities.

Chapter 5

Methodology

In this chapter, the proposed method in this work will be explained. Firstly, the implemented network will be illustrated. Secondly, the implementation details and my amelioration process will be explained.

5.1 Proposed Model

The aim is to create a model adapted for the task of Twitter document representation that leverages multilingual and multimodal features to get a better performance. For this end, the idea was to adapt the architecture of the Se-MLNN proposed by [CHMBE21] (shown in Figure 5.1), and implement additional parts that were not taken into consideration in the previous work. I used the same structure of layers to investigate the added value of different visual features in combination with contextual language representations for multimodal multilingual tweet classification. Also, a concatenation module was adopted as a primary solution to merge the text and visual features after passing both of them through the multi-layer neural network. As opposite to the work proposed by [CHMBE21], instead of investigating the impact of several high-level visual features such as object, places, and facial, only object features were considered.

The proposed model shown in Figure 5.2 has three global inputs: tweet, quotes and replies. Every tweet, quote, and reply is accompanied by its corresponding image if it is available. A vector filled with zeros is inputted if no image was detected. After investigation of the

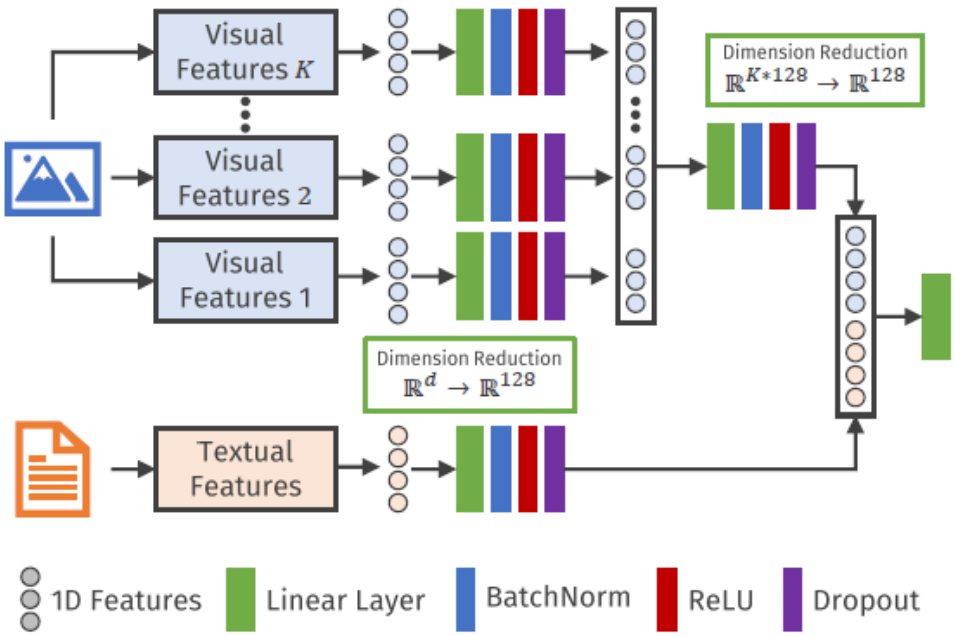


Figure 5.1: Se-MLNN: Proposed architecture for multimodal sentiment classification [CHMBE21].

given data, each document can have a number of quotes and replies between 0 and 867. Every tweet T is composed of the pair (t, v) where t represents the textual part and v refers to the visual part of the respective piece of input passed through a main branch in the proposed model. I have three main branches, one for each main input (tweets, replies, and quotes). Each main branch is composed of two parallel branches, one for each part of the pair (t, v). Each branch enables the model to learn the representation of each modality. Both language and visual embeddings are combined to obtain a multimodal representation for each main/global branch. The final fusion of different VSE is then held to get the document representation.

A fusion of the different branches based on attention mechanisms is used to capture the common inter-modal and unique intra-modal [ZLY21] information for multimodal document analysis. In fact, at the level of each main branch of replies and quotes, each reply and quote has its own unique characteristics. These characteristics can be for example a significant word or part from an image. For example, the following sentence is given: "She is eating a green apple". In this case, the most relevant words are "eating" and "apple" because they contribute the most to the overall meaning of the sentence. So a self attention layer [ZLY21]

is placed to know which features better contributes to the multimodal representation of that specific reply or quote. On the other hand, another attention mechanism that takes the tweet as a query and each reply and quote as context [VSP+17] is used to enhance meaningfulness of the overall document representation. In this context, more weight is given to a reply, for example that contribute better to the context of the tweet.

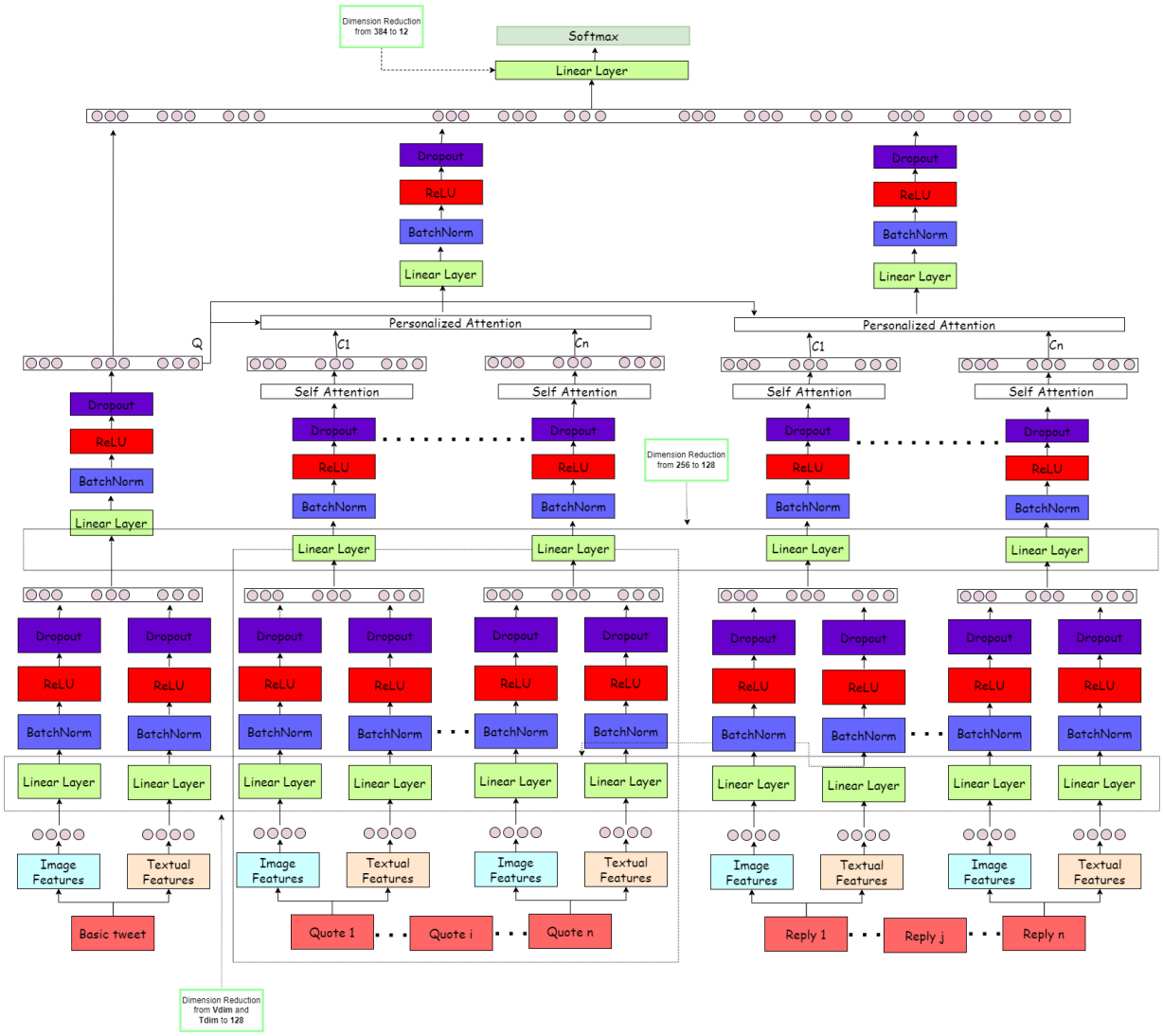


Figure 5.2: HBMnet: Architecture for multilingual multimodal document representation.

The main differences between my model called HBMnet and the one proposed in [CHMBE21] called Se-MLNN lies at three levels: the number of elements within a document, the used method to combine different modalities and the used language. In Se-MLNN, two main

components were used as inputs: One image and one tweet text per document. In HBMnet, I extend this approach by adding replies and quotes texts to each document composed of a main tweet text. All these semantic features comes with their corresponding visual features if available. Fusing different modalities in Se-MLNN is simple, as visual and textual features are concatenated. In HBMnet on the other hand, several techniques of fusion based on attention mechanisms were used to investigate the impact of intra-modal and inter-modal information on the overall document representation.

5.2 Multilingual and Multimodal Model Implementation

In this section, different architecture decisions as well as the reasons that each one is utilized are discussed. For each module that is used, the theoretical foundation and the implementation decisions are analyzed. The goal of this thesis is to learn tweet representations based on multilingual and multimodal features and classify them into several topics. To this end, modules that extract multilingual and multimodal features are firstly presented.

Finally, I debate the changes I considered, improving the final results.

5.2.1 Multilingual and Multimodal tweet representation

At the level of extracting vision features, I follow the work of [CHMBE21] by making use of ResNet-50 [HZRS16] and its last convolution layer in order to extract vision object features instead of the last layer that presents the object categories. Here I get an output of 2048 feature maps each of size 7×7, which will enable to get 2048-dimensional vector after pooling with a global average.

At the level of extracting semantic features, I presume that the meaning of words within one tweet sentence are equally important to contribute to the final tweet embedding. To this end, I extract contextual word embeddings using XLM-Roberta and employ different pooling strategies to get a single embedding for the tweet. I follow the experimental founding presented in [CHMBE21] and calculated the average of the last four layers, which were proved to be the most useful. I finally take the word embeddings and calculate the average

in order to get a single tweet text embedding of 768 dimensions. The text embedding problem for one single tweet can be formulated as follows:

$$\text{sentence embedding} = \frac{1}{n} \times \sum_{i=1}^{n} \text{token}_i \tag{5.1}$$

where n is the number of tokens in a single sentence representing a tweet, reply or quote and the token is the averaged vector of the last four layers with a length of 768.

I also added the number of likes for each quote and reply as a feature to further improve the final representation of the document. The idea behind this is that the most significant replies and quotes will be associated with a bigger number of likes by the Twitter users. When a given reply for example contributes better to the idea of the tweet and this reply is associated with a big number of likes, in this case the model will learn this relation between the two parts.

I used for this the same architecture as for adding a reply or a quote. In this case, the vision, language, and number of likes features get fused together at each branch before fusing all parts of each document. the architecture of this added feature is shown in Figure 5.3.

5.2.2 Tweet Preprocessing

A number of steps are taken in order to clean the textual data from unnecessary or noisy strings such as the removal of non-necessary white spaces, links, and line breaks in order to create an appropriate vector representations of each tweet text. I make use of the text processing tool developed in [BPD17] which can perform social-aware (understands complex emoticons, emojis, and other unstructured expressions like dates, times), tokenization, spell correction, word normalization, word segmentation and word annotation. This tool will help in the understanding of different structures of the given text as it was trained over a large corpus of 330 millions tweets. Unfortunately, not all of these features are used in this project due to its limitations, such as the spell corrector which does not support multilingual language input. That is why I restricted the utilization of this tool along with its tokenizer to identify emojis and expressions. These expressions can be for example dates (22/08/2022, April 23rd), times (4:30 pm, 11:00 am) and currencies (10$, 25mil, 50€).

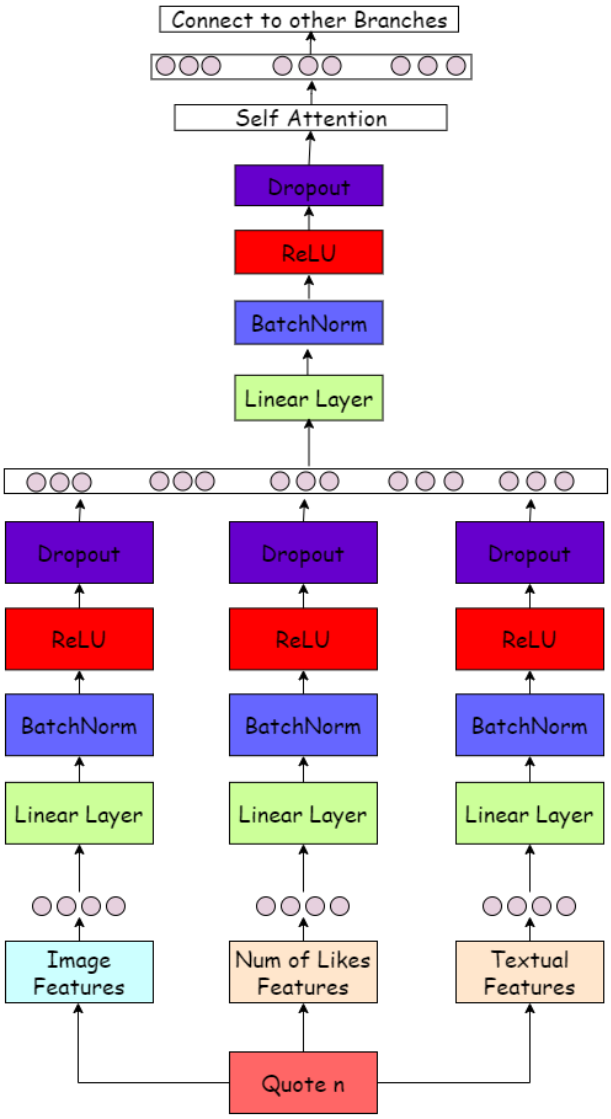


Figure 5.3: Quotes branch architecture from HBMnet's model. Number of likes feature added similarly as textual and image features.

After the tokenization, an extra preprocessing phase is added. In fact, URLs and mentions (a user notifying another user in the text with the "@" symbol) are removed using regular expressions. Hashtags (using a topic related keyword with the "#" symbol like #Climate-Change) and line breaks are cut out in the same way. As the hashtags are used as classes in the following classification tasks, they should be removed because that is the model's main task to predict. More details on how hashtags were exploited is presented in the next section. Moreover, words reserved from Twitter, such as RT and VIA, are removed and

replaced with one single space respectively. An example of the text preprocessing on a Twitter message is shown in Table 5.1.

Table 5.1: Example of tweet text preprocessing procedure

Original text	The *new* season of #TwinPeaks is coming on May 21, 2017. CANT WAIT \o/ !!! # tvseries #davidlynch :D
Processed result	Tthe new <emphasis> season of twin peaks is coming on <date> . cant <allcaps> wait <allcaps> <happy> ! <repeated> tv series david lynch <laugh>

5.2.3 Multimodal Feature Fusion

Inspired by the work in [YYS+21], I first used a very simple technique which is concatenating the vision and the language features of each tweet, reply, and quote and fed them into fully connected neural network layers for cross-modality feature learning. I repeat this procedure between all replies and all quotes so that I get at the end a final embedding that represents the hole document which I can classify. This approach proved to have a good performance [HF18].

To further enhance the performance of my model, I make use of attention based mechanisms. This allows the model to focus on the most significant features in quotes and replies and give it more weights. Following the work in [DLA17], [ZLY21], [ZZS+19], [WWA+19] and [TL19] where different strategies of attention based solutions were proposed, I leveraged the idea of Scaled Dot-Product Attention which architecture is shown in Figure 5.4.

The proposed attention is based on (Q, K, V) concept with Q: Query, K: Key and V: Value. As shown in Figure 5.2, I use the tweet, each reply and quote as inputs to the attention module. In this case, I define the tweet as the query, the keys, and the values as the VSE outputted from each corresponding reply and quote. To understand why K and V are based on similar inputs, let's take the following example: When searching for videos on YouTube, the search engine will map your query (text in the search bar) against a set of keys such as video title, description associated with candidate videos in their database, then present you the best matched videos (values). In this thesis, the model should choose among the VSE of different replies and quotes which one contribute the most to the idea of the tweet. In practice, the computation of the attention function is held simultaneously on a set of

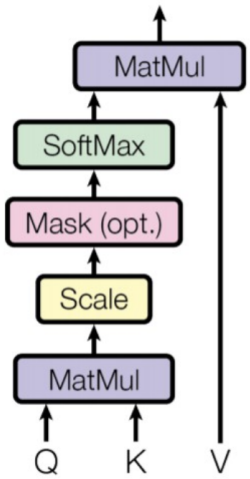


Figure 5.4: Attention Mechanism with Scaled Dot-Product Attention [VSP+17]

queries that are packed together into a matrix Q. Every query includes the pair (tweet, reply) or (tweet, quote). The attention function in this case can be formulated as follows:

$$\text{Attention}(Q, K, V) = Softmax(QK^T)V \tag{5.2}$$

It is also to mention, that the proposed model accepts a fixed number of replies and quotes, but not every tweet comes with that number of interactions relevant to it. in that case, I will get matrices filled with zeros. If, for example, I fix the number of replies to 100 and for a tweet that has only five corresponding replies, I will get 95 other vectors that are filled with zeros. In this case, the effect of these available replies will be diluted when the weights are calculated. To solve this issue, I used the masking implementation in [VSP+17] and thus by setting to minus infinity all the values in the inputs of the Softmax which correspond to illegal connections. Illegal connections here represents replies that have just zeros in their matrices.

Also, to improve the representation within each modality, i.e., quote and reply, a self attention module was used at each branch. Based on the principle of functionality of Bert [DCLT18] i.e., how it can understand each word based on its context, I use self attention as a way to relate different parts of a single sequence in order to learn a better representation of the sequence. In this work, the sequence that is given as input is the combination of the

visual and semantic features of each quote and reply.

For both attention modules, I use the implementation of PyTorch NLP [1] by getting the mix variable concatenated with query, passed through a linear layer and a tanh activation function afterwards.

5.3 Multi-class and Multi-label Classification

In this part, I present two approaches that were considered to classify the tweets. I approached the classification task problem in a Multi-class and Multi-label manner.

5.3.1 Multi-label

After building my topics based on hashtags like explained in subsection (4.2), it is to notice that some tweets belongs to two or more topics. The Table 5.2 shows the number of tweets associated with each topic.

Responding to the multi-label classification task, I use the BCEWithLogitsLoss [2] loss function [LZZP22], which combines a Sigmoid layer and the binary cross-entropy loss (BCELoss) in one single layer. As a result, it is more stable numerically than a plain Sigmoid layer followed by a BCELoss. By combining the operations into one layer, one takes advantage of the log-sum-exp trick [NS16] for numerical stability. In practice, I get an output for each class for a given document. If the value is greater or equal than a threshold, that class label is assigned to that document (More than one label could be assigned to a given document).

The calculation of this loss can be formulated in the equation 5.3:

$$l = -[y \cdot log(\delta(x)) + (1 - y) \cdot log(1 - \delta(x))] \tag{5.3}$$

Where x is the input without sigmoid function, y is the corresponding ground truth, and $\delta(x)$ is the sigmoid function, which is used to map the input x to the interval of [0,1]. The calculation of the sigmoid function is formulated in the equation 5.4:

$$\delta(x) = \frac{1}{1 + \exp(-x)} \tag{5.4}$$

[1] `https://pytorchnlp.readthedocs.io/en/latest/index.html`

[2] `https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html`

Table 5.2: Number of tweets within each topic in multi-label classification

Topic	Number of tweets
automotive Industry	367
automotive Industry, economics, technology	180
automotive Industry, fun	6
automotive Industry, fun, economics, technology	38
corona	4011
economics	1269
fun	2049
fun, promotion	300
history	605
politics	4504
politics, economics	102
politics, economics, promotion, social	59
politics, promotion, social	2909
promotion	928
sport	8947
war, politics	8925
war, politics, promotion, social	42
weather	808
Sum	**36049**

5.3.2 Multi-class

To answer to the proposed classification task, I also used a multi-class manner. The reason behind that is that some topics does not have enough training data instances to learn the relation between the document features and the topics. An example of this is found in Table 5.2 of the topic "automotive industry, fun" which has only 6 instances. To this end, I eliminate all topics that includes more than one label in order to get 12 topics as shown in Figure 5.5.

In the case of multi-class classification, I use CrossEntropyLoss [3] loss function, which is

[3] `https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html`

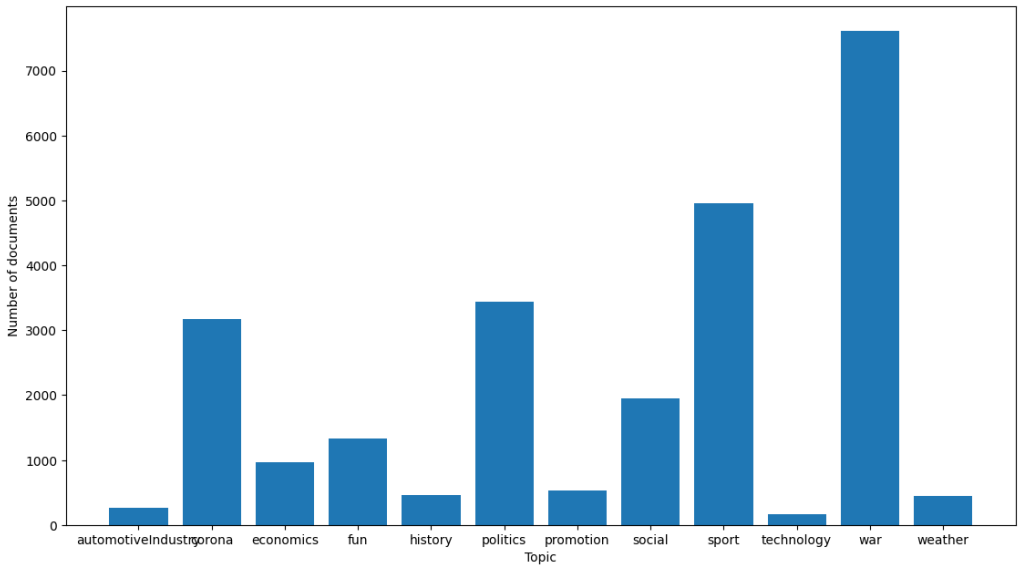


Figure 5.5: Distribution of the number of tweets per topic with multi-class classification

adapted to multi-class classification problems with a defined number of classes and returns a probability value between 0 and 1 for each class. The sum of probabilities of all classes should be 1. The calculation of this loss can be formulated as follows:

$$l = -\log \frac{\exp(x_y)}{\sum_{i=1}^{c} \exp(x_i)} \tag{5.5}$$

where x is the input vector of size c, c is the number of classes and the target $y \in [0, c)$

Chapter 6

Experiments and Results

In this chapter, the process that is used to evaluate the performance of the proposed architecture is described. First, I present the results of the baselines. I then evaluate the performance of the added features and module, i.e., interactions to the tweet, number of likes feature and the attention mechanisms. Furthermore, I show the results of combining all these parts. Finally, I examine the Multilingual Multimodal Document Representation.

All the implementations of the previously described modules are done using Python language. In particular, for the purposes of implementing Neural Networks, PyTorch[PGM+19] framework is utilized. scikit-learn [PVG+11] was used for metrics computation purposes. Moreover, both Pandas [pdt20] and Numpy [HMvdW+20] libraries for python are employed for data handling and basic manipulations. Moving forward, I present the used parameters setting.

6.1 Parameters Setting

In all experiments, the training for all models was using an early stopping configuration. The maximum set number of epochs is 1000, but the training will stop if the performance is not increased for the last 20 epochs, i.e., the learning process will stop automatically if the model decides that nothing more could be learned. I use as well an Adam optimizer for updating the neural network parameters. The scheduler of the optimizer will read the metrics quantity and if no improvement is detected after a fixed number of epochs equal

to 5, the learning rate is reduced. The learning rate is initially set to, 1×10^{-4} and the updated learning rate will be multiplied by a factor of 0.1 if the validation loss does not decrease. A batch size of 128 is used. Like specified in [CHMBE21] a ratio of 0.5 for the dropout is applied after all the intermediate linear layers to avoid over-fitting.

All models were trained on one GPU (NVIDIA GeForce GTX 1080ti with 12 GB Memory) with eight workers.

I conduct 5-fold cross-validation with a split of 0.765, 0.135 and 0.1 for training, validation, and test sets respectively. During this process, I used a stratification technique to ensure that the data samples are well divided, i.e., each split has data samples from each class.

6.2 Metrics

In this section, the metrics used to evaluate the trained models in the experiments will be explained.

I first calculated the **accuracy score** for both multi-label and multi-class classification tasks. In multi-label classification, the accuracy score is calculated based on whether all ground truth labels are predicted correctly. In this case, The set of labels that were predicted for a given sample must exactly match the corresponding set of labels in ground truth. In multi-class classification, the same function is used, but ground truth will just set to be including one label. The computation of the accuracy score can be formulated as follows, in the equation 6.1:

$$Accuracy\ Score = (TP + TN)/(TP + FN + TN + FP) \tag{6.1}$$

With TP representing true positives, TN representing true negatives, FN representing false negatives and FP representing false negatives.

I also calculated **Weighted F1 score** [1] for both above cited tasks. The formula to this metric can be written as follows in the equation 6.2:

$$F1 = \frac{2 \times (precision \times recall)}{(precision + recall)} \tag{6.2}$$

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html`

Moreover, I calculated a **confusion matrix** to be able to evaluate the performance of the classification algorithm. This will help to get an overview about the number of observations known to be in topic i and predicted to be in topic j.

6.3 Initial Document Classification

Similar to the architecture of the Se-MLNN [CHMBE21], two baselines are proposed. The first one called **hbmnet_baseline_t** takes into consideration only one text modality, which is the tweet. The purpose of this baseline is to provide a point of reference for other developed models. Also, it was used to show the added value of the other modalities in the final document representation. The second baseline called **hbmnet_baseline_tv** is almost identical to Se-MLNN proposed by [CHMBE21]. The main difference is that in my proposed baseline called **hbmnet_baseline_tv** handles the absence of an image in the pair (t,v). The results of the initial document classification are shown in Table 6.1

Table 6.1: Baseline unimodal and bimodal feature results for MCMTRA_1 and MCMTRA_2. Accuracy and F1 scores are averaged over 5 folds. The best results for each subdataset and method are written in bold.

Method	Model	MCMTRA_1		MCMTRA_2	
		ACC	F1	ACC	F1
Multi-label	hbmnet_baseline_t	62.13	69.73	61.83	70.33
	hbmnet_baseline_tv	**63.95**	**70.22**	**62.12**	**70.62**
Multi-class	hbmnet_baseline_t	64.52	68.82	63.83	74.53
	hbmnet_baseline_tv	**64.67**	**69.02**	**64.12**	**74.56**

It is to notice that the validation accuracy results for MCMTRA_1 are overall better than for MCMTRA_2. This justifies the important rule of the corpus size: The more data samples are available, the better the results will be. If I compare the results issued from both methods, I found that the multi-class gives slightly better performance. That proves the hypothesis explained in Chapter 5 about the reasons on why should I eliminate classes

with low data samples. Furthermore, the bimodal baseline **hbmnet_baseline_tv** improved slightly the results for both subsets, which proves the importance of adding visual features to model.

Even though I run several experiments with the 3 versions of XLM-RoBERTa like explained in subsection 2.5, reporting results for all these combinations is not informative because the fine-tuned version Twitter-xlm-roberta-base proved to be the best performing language pre-trained model in most of the experiments.

I also tested the baseline hbmnet_baseline_tv using vision pre-trained model ResNet-50 for object detection and ResNet-101 for place and scene features detection. These experiments proved that ResNet-50 is the most effective vision pre-trained model for this task.

In the next sections, I only use the embeddings issued from the best language and vision pre-trained models, Twitter-xlm-roberta-base and ResNet-50 respectively. Moving forward, I present the experiments for multimodal models and their corresponding results.

6.4 Multimodal Document Classification

Having analyzed the results of the baselines, now I continue by proposing a data imbalance handling process and discussing the experiments held to evaluate the extracted multimodal features. First, every model with different added modules and features i.e., interactions to the tweet, number of likes feature and the attention mechanisms, is tested separately, reporting their results and naming the best one of each module, while later, different combinations of modules are attempted and those that increase the metrics are kept.

6.4.1 Data imbalance Handling

Moving from multi-label classification task to the multi-class classification task reduced the number of classes by removing those who have more than one label as ground truth. Also, the use of a stratification module has assisted the distribution of data samples of each class among the different splits. Even though the solutions cited above were conducted, the experiments in the last section show that an imbalance in the presented dataset still persist. I remark that after running one epoch of the model **hbmnet** the diagonal of the confusion matrix was filled with zeros for classes that does not have enough data samples, which

means that the model did not have the chance to learn any feature from these classes. To this end, I opted for the weighted random sampler provided by PyTorch, which generates weights for each class based on the supplied number of samples. In this case, if I take an example of the class "war" having 3000 data samples and a class "corona" having only 200 data samples, then the weights passed to the train data loader will be $\frac{1}{3000}$ and $\frac{1}{200}$ respectively.

6.4.2 Impact of Multimodal Features on Document Classification

In this section, I investigate the impact of different components on the overall representation of tweets. I begin with the **hbmnet** model, which has the same architecture as the proposed method presented in Figure 5.2 but without considering the self-attention modules that were added at each branch and the attention module between the tweet and their corresponding interactions. In this case, inspired by [CHMBE21] a direct concatenation between different features is held. After that, I tested the added value of number of likes for each reply and quote within each document in **hbmnet_w_likes**. I have also run two experiments to investigate the improvement of the implemented attention mechanism: **hbmnet_w_self_attention** for the self-attention at each branch of each reply and quote and **hbmnet_w_attention** for detecting the relation between the tweet and each of the other interactions.

For all these experiments, I use MCMTRA_1 and MCMTRA_2 subdatasets and I opted for the multi-class classification task. The results of these experiments are shown in Table 6.2. Overall, the results for the subdataset MCMTRA_2 where the document include tweets with at least two replies and one quote shows better performance in comparison to the subdataset MCMTRA_1 where no constraints on the document were applied. This proves the importance of the availability of replies and quotes within each document. Adding the number of likes feature do not show a significant improvement with MCMTRA_2 subdataset and decreased the performance with the use of the MCMTRA_1 subdataset. Adding the self-attention modules on the quotes and replies branches and attention between the tweet, replies and quotes increased the performance by 1% to 3% in comparison to the basic concatenation based module adopted in **hbmnet**.

Table 6.2: Baseline unimodal and bimodal feature results for MCMTRA_1 and MCMTRA_2. Accuracy and F1 scores are averaged over 5 folds. The best results for each subdataset and method are written in bold.

Method	Model	MCMTRA_1		MCMTRA_2	
		ACC	F1	ACC	F1
Multi-class	hbmnet	65.13	68.63	66.23	69.81
	hbmnet_w_likes	64.23	67.35	66.26	69.36
	hbmnet_w_self_attention	66.44	**69.85**	68.76	**70.76**
	hbmnet_w_attention	**66.65**	69.56	**68.89**	70.36

6.4.3 Results Combination

Now that the results for individual modules have been presented, the experiments conducted for different combinations of these modules are shown in this section. I want to test the combination of the most promising features/modules to further improve the performance. in Table 6.3, two additional experiments are carried on, to understand what is the most prominent combination. First, I combine the attention module between the tweet and the replies and quotes and self attention with the number of likes feature. Second, I combine both attention based systems without adding the number of likes features. The results in Table 6.2 showed that having a subset with more constraints, i.e., a tweet having more replies and quotes, will result in a better performance of the model. That is why I only use MCMTRA_2 subdataset in a multi-class classification manner for these experiments.

Table 6.3: HBMnet: combining self-attention with attention and likes with attention feature results for MCMTRA_2. Accuracy and F1 scores are averaged over 5 folds. The best results for each dataset are written in bold.

Subdataset	Model	ACC	F1
MCMTRA_2	hbmnet_attention_likes	68.33	70.09
	hbmnet_w_double_attention	**70.42**	**73.46**

The table 6.3 shows that adding number of likes feature along with the attention mechanism decreases the performance. It shows even less performing results in comparison to **hbmnet_w_self_attention** and **hbmnet_w_attention** described in Table 6.2. These results show that the proposed model get affected by this extra information. The combination of the proposed attention mechanisms on the other hand considerably increased the performance by at least 4% in comparison to **hbmnet** model. For a better overview, The Table 6.4 shows all used models along their corresponding description.

Table 6.4: Description of different used models.

Model	Description
hbmnet_baseline_t	multilingual baseline model for text
hbmnet_baseline_tv	multilingual baseline model for text and image
hbmnet	multimodal multilingual model with replies and quotes
hbmnet_w_likes	same as "hbmnet" with number of likes feature
hbmnet_w_self_attention	same as "hbmnet" with self attention module
hbmnet_w_attention	same as "hbmnet" with attention between tweet and interactions
hbmnet_attention_likes	same as "hbmnet" with both attention mechanisms and number of likes feature
hbmnet_w_double_attention	same as "hbmnet" with both attention mechanisms

To further enrich the evaluation process, I make use of confusion matrices. The Figure 6.1 shows the confusion matrix of best proposed model **hbmnet_w_double_attention** after averaging the results over 5 folds. The results show that not all classes are being learned as they are supposed to. Although, an imbalance handling module as described in section 6.4.1 was implemented, this hasn't improved significantly the results for all classes, such as for the case of class "technology" whose documents were also predicted with important percentages as elements of the classes "economics" and "social". This suggests that the documents belonging to each of these topics have similar content.

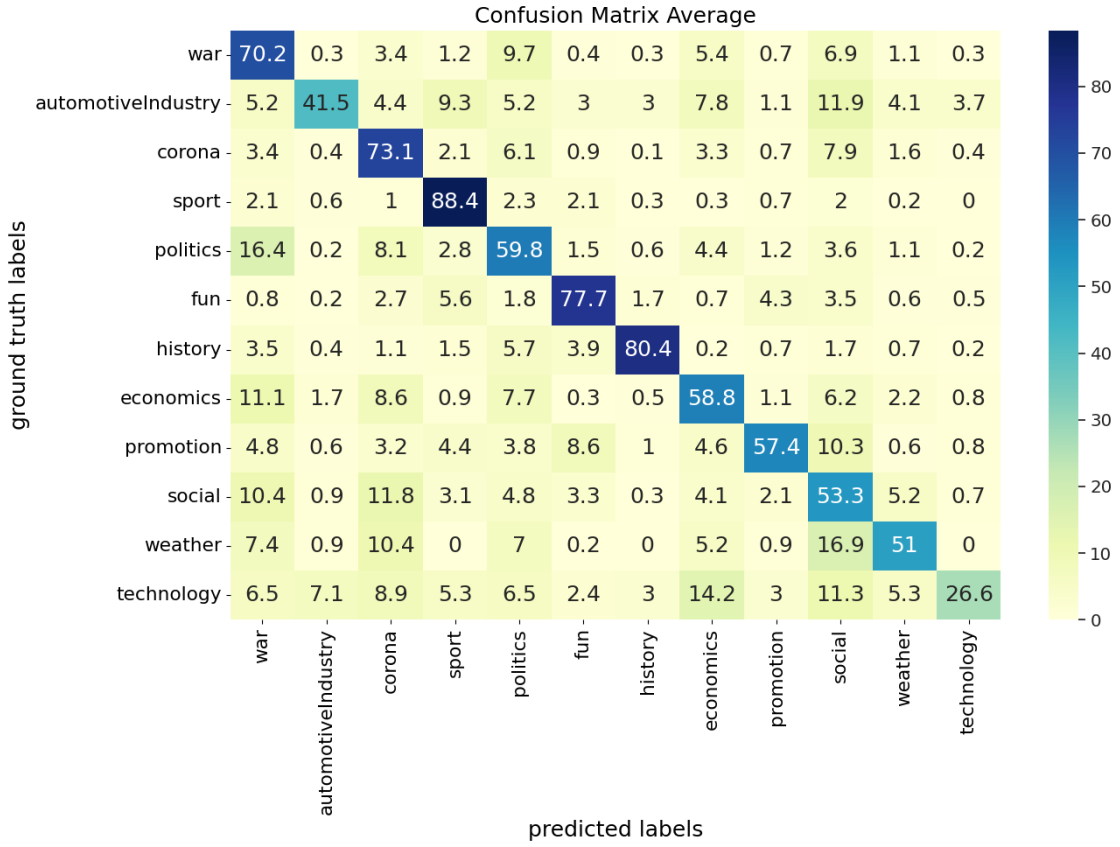


Figure 6.1: Confusion matrix for the proposed model. The results are averaged over 5 folds

6.5 Multilingual document classification

To evaluate the performance of the proposed model for the multilingual document classification task, I train the model with only documents from one language and then test the performance of that model on other languages. To this end, I choose a subset that only includes documents in German language and are assigned to one of the following three topics(”corona”, ”politics” and ”sport”). I opted for this choice, because these are the only classes/language for which enough data was collected to execute the learning process of the model. The subset includes 13,460 tweets, 25,559 replies and 7,690 quotes along with their corresponding images if available. I then test the model on two subsets, one for English language and one for French language. The English based subset includes, 3,635 tweets, 4,083 replies and 2,698 quotes. The French based subset includes, 10,334 tweets, 14,442 replies and 10,695 quotes. The results of the validation and testing process are presented

in Table 6.5. The results show that the performance for the validation set on German language and the testing set on French language are very close to each other. This proves that embeddings issued from both these languages are more aligned. On the other hand, a noticeable decrease in performance is detected for testing on English language. This suggests that the embeddings issued from the German/English languages are less aligned.

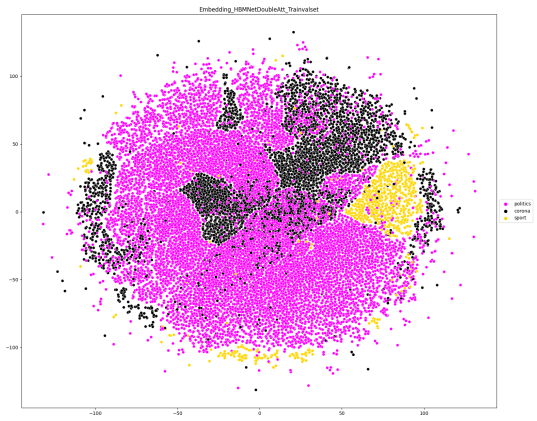
Table 6.5: Multilingual validation and test of HBMnet on German, French, and English languages. Best results for accuracy and F1 scores are presented.

Metric	Valset German	Testset French	Testset English
Accuracy	86.09	85.64	73.64
F1	86.22	85.02	75.41

To gain insights on the tweet embeddings, I visualize the high-dimensional embedding vectors using t-SNE [VdMH08] projections. This will give each high-dimensional data point a location in a two-dimensional space, along with preserving the relative distances between the embedding vectors. Figure 6.2 shows that even though the model did not use any parallel German-French-English data for the training process, it still managed to produce aligned embeddings across these three languages. This is proved in Sub-figure 6.2c and 6.2d where the separation between the three classes can be well observed. The Sub-figures 6.2a and 6.2b show the high-dimensional embedding vectors for the training and validation process in German language. The Sub-figure 6.2a presents a good quality of training, as the three classes can be easily distinguished. The sub-figure 6.2b confirm the results of the training, as the model was able to separate the three classes clearly.

6.6 Multilingual Multimodal Document Representation

To examine the high-level document representations learned by the proposed model, I extract the features outputted from concatenation of the three branches of the proposed architecture **HBMnet** presented in Figure 5.2. I use the best performing trained model over five folds. I also follow the same method for the bimodal baseline model. I do a forward pass using the data instances that belongs to the validation set issued from the



(a) Trainset with German language

(b) Valset with German language

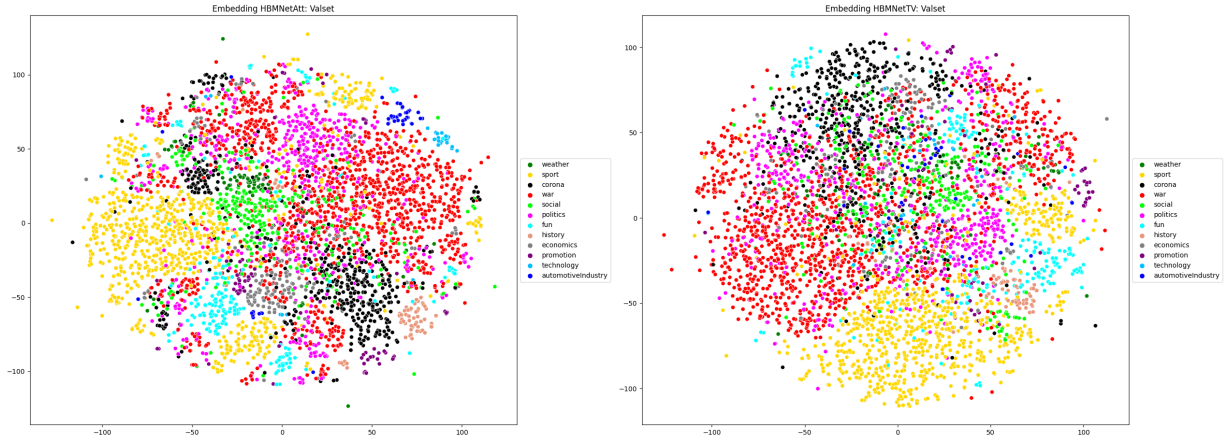
(c) Test with English language

(d) Test with French language

Figure 6.2: T-SNE visualization for trainset, valset and testsets. The trainset and valset are issued from German language. The testsets are issued from French and English language.

MCMTRA_2 subset, leaving as output the embedding vector for each document. I then perform t-SNE [VdMH08] analysis on these points, shown in Figure 6.3.

I observe that the points presented in the proposed model shown in the Sub-figure 6.3a form better clusters than the ones presented in Sub-figure 6.3b. This is relevant if the topic "corona" shown in black is considered. This proves that, the top-level embeddings issued from the proposed architecture carry information which is more discriminative across



(a) t-SNE visualization for valset with the proposed model

(b) t-SNE visualization for valset with the bimodal baseline

Figure 6.3: T-SNE visualization of validation set for the bimodal baseline and the proposed model.

different topics. This supports the hypothesis that attention mechanism and the use of multi-modalities i.e., replies and quotes can enhance the learning with more plausible embeddings into a shared semantic space across multiple languages.

Chapter 7

Conclusion

The problem of data representation learning in Twitter was addressed in this thesis. This task is particularly challenging due to the rarity of multilingual multimodal annotated datasets. Also, although several research works addressed different tasks in the context of multilingual and multimodal data analysis, like sentiment analysis and topic classification, most of them concentrate on one aspect. The duality of these two aspects, i.e., multilingual and multimodal, has been employed and tested in this work.

In this context, a novel Deep Learning based data representation learning method by leveraging a topic classification task is proposed. This method has taken a step further to reinforce the importance of combining tweet text features and corresponding interactions such as replies, quotes, and images to represent the data in a more accurate way. On the other hand, it has suggested that adding number of likes feature doesn't improve the learning process.

To assess the effectiveness of the proposed approach, a series of experiments were carried out. Different subsets of the dataset "MCMTRA" were used for training, evaluation, and testing. The results show that even having less data instances, more interactions related to each tweet is a very important factor to represent data in a more accurate way. It has also been shown that adding attention mechanisms enhance the performance of the model to learn features that contribute more to the final representation of the document. Moreover, the proposed model proved to be well performing for the multilingual task, as most languages used in this work had aligned embeddings. Overall, the model that this

work proposes is based on the usage of tweets, replies, and quotes along their corresponding images if available, attention mechanisms and dataset that has as many interactions as possible within each document as it yields the best scores in the reported metrics and showed better representations.

Limitations in the current work are present at two levels: in the collected data and the model. In fact, not all documents have enough interactions, i.e., replies and quotes and images, which will prohibit the model from learning all document representations equally. Also, the process of labeling the dataset is done using a first manual inspection, which is very costly. Moreover, the dataset was not balanced as some topics includes less than 1000 documents, which limited the prediction capacity of the model for these topics. The model also learns from noisy labels extracted from hashtags. Furthermore, the model performs poorly on under-represented classes, although I used a weighted sampling mechanism to mitigate the data imbalance. Future work could hence explore other techniques to address this problem, like weighted loss. In addition, the model's parameters were empirically set due to time limitation.

Taking into account the results of the current work and their respective limitations, I propose some directions for future research that will address the problem of data representation learning with Tweet data using a multimodal multilingual model. Labelling the data with hashtags can be improved by developing a more sophisticated manner based on machine learning to understand the relation between the hashtags and classify them into topics. This can be performed by extracting embedding of hashtags and learn a representation that can enable a more accurate classification of the labels [LHH18]. Another perspective to perform a classification task and reduce the imbalance of data problem among topics would be to perform another training task that concentrate the work on detecting a certain pattern, like sarcasm detection. In addition, other techniques for data imbalance could be explored to address this problem, like weighted loss. Moreover, different preprocessing steps on Tweet text can be added, such as multilingual spell correction. Future work could also explore automatic hyperparameter tuning, e.g., using grid search.

Data representation for Twitter is one of the highly demanded topics among research communities. This domain is infant when working in multilingual and multimodal features, and has great potential to advance the ecosystems in which millions of people spend much of their time daily.

List of Figures

List of Tables

Bibliography

[AAJ+21] Luqman Ali, Fady Alnajjar, Hamad Al Jassmi, Munkhjargal Gocho, Wasif Khan, and M Adel Serhani. Performance evaluation of deep cnn-based crack detection and localization techniques for concrete structures. *Sensors*, 21(5):1688, 2021.

[AAL+15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[AHB+18] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[Ang20] Dimo Angelov. Top2vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470*, 2020.

[B+95] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[BCB14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[BEACC22] Francesco Barbieri, L Espinosa-Anke, and Jose Camacho-Collados. Xlm-t: Multilingual language models in twitter for sentiment analysis and beyond.

Proceedings of the LREC, Marseille, France, pages 20–25, 2022.

[Bho20] Ayan Kumar Bhowmick. Temporal pattern of retweet (s) help to maximize information diffusion in twitter. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 913–914, 2020.

[BPD17] Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 747–754, 2017.

[CCDMC+20] Jose Camacho-Collados, Yerai Doval, Eugenio Martínez-Cámara, Luis Espinosa-Anke, Francesco Barbieri, and Steven Schockaert. Learning cross-lingual word embeddings from twitter via distant supervision. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 72–82, 2020.

[CCW19] Yitao Cai, Huiyu Cai, and Xiaojun Wan. Multi-modal sarcasm detection in twitter with hierarchical fusion model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2506–2515, 2019.

[CDFT] Stefano Cresci, Felice Dell'Orletta, Fabrizio Falchi, and Maurizio Tesconi. Cross-media learning for image sentiment analysis in the wild.

[CHMBE21] Gullal S Cheema, Sherzod Hakimov, Eric Müller-Budack, and Ralph Ewerth. A fair and comprehensive comparison of multimodal tweet sentiment analysis methods. In *Proceedings of the 2021 Workshop on Multi-Modal Pre-Training for Multimedia Understanding*, pages 37–45, 2021.

[CKG+19] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

[CLC17] Iacer Calixto, Qun Liu, and Nick Campbell. Multilingual multi-modal embeddings for natural language processing. *arXiv preprint arXiv:1702.01101*, 2017.

[CLY+20] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[DGG+21] Erwan Dujeancourt, Marcel Garz, Anindya Ghose, Johannes Hagen, Juliane Lischka, Mattias Nordin, Jonna Rickardsson, and Marco Schwarz. The effects of algorithmic content selection on user engagement with news on twitter. Technical report, Working Paper, 2021.

[DL17] Bo Dai and Dahua Lin. Contrastive learning for image captioning. *Advances in Neural Information Processing Systems*, 30, 2017.

[DLA17] Chi Thang Duong, Remi Lebret, and Karl Aberer. Multimodal classification for analysing social media. *arXiv preprint arXiv:1708.02099*, 2017.

[DPH15] Nugroho Dwi Prasetyo and Claudia Hauff. Twitter-based election prediction in the developing world. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, pages 149–158, 2015.

[EFSS16] Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. Multi30k: Multilingual english-german image descriptions. *arXiv preprint arXiv:1605.00459*, 2016.

[Fai20] Dinar Faiza. Stimulating english learning in global kpop community on twitter. *Journal of Applied Linguistics (ALTICS)*, 2(1), 2020.

[Fel10] Christiane Fellbaum. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer, 2010.

[Fer14] Susanne Fertmann. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. 2014.

[FFKF17] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*, 2017.

[GD17] Dagmar Gromann and Thierry Declerck. Hashtag processing for enhanced clustering of tweets. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 277–283, 2017.

[GEM16] Rehab S Ghaly, Emad Elabd, and Mostafa Abdelazim Mostafa. Tweets classification, hashtags suggestion and tweets linking in social semantic web. In *2016 SAI Computing Conference (SAI)*, pages 1140–1146. IEEE, 2016.

[GKSS+17] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

[GSKL17] Spandana Gella, Rico Sennrich, Frank Keller, and Mirella Lapata. Image pivoting for learning multilingual multimodal representations. *arXiv preprint arXiv:1707.07601*, 2017.

[GSZ+16] Saadiah Ghazali, Nor Intan Saniah Sulaiman, Nerda Zura Zabidi, Mohd Faizal Omar, and Rose Alinda Alias. The impact of knowledge sharing through social media among academia. In *AIP Conference Proceedings*, volume 1782, page 030003. AIP Publishing LLC, 2016.

[GWCC17] Jiuxiang Gu, Gang Wang, Jianfei Cai, and Tsuhan Chen. An empirical study of language cnn for image captioning. In *Proceedings of the IEEE international conference on computer vision*, pages 1222–1231, 2017.

[HAR14] Maryam Hasan, Emmanuel Agu, and Elke Rundensteiner. Using hashtags as labels for supervised learning of emotions in twitter messages. In *ACM SIGKDD workshop on health informatics, New York, USA*, volume 34, page 100, 2014.

[HCC+19] Eduard H Hovy, Jaime G Carbonell, Hans Chalupsky, Anatole Gershman, Alex Hauptmann, Florian Metze, Teruko Mitamura, Zaid Sheikh, Ankit

Dangi, Aditi Chaudhary, et al. Opera: Operations-oriented probabilistic extraction, reasoning, and analysis. In *TAC*, 2019.

[HCHH20] Po-Yao Huang, Xiaojun Chang, Alexander Hauptmann, and Eduard Hovy. Forward and backward multimodal nmt for improved monolingual and multilingual cross-modal retrieval. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pages 53–62, 2020.

[HF18] Anthony Hu and Seth Flaxman. Multimodal sentiment analysis to explore the structure of emotions. In *proceedings of the 24th ACM SIGKDD international conference on Knowledge Discovery & Data Mining*, pages 350–358, 2018.

[HKL+19] Po-Yao Huang, Guoliang Kang, Wenhe Liu, Xiaojun Chang, and Alexander G Hauptmann. Annotation efficient cross-modal retrieval with adversarial attentive alignment. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1758–1767, 2019.

[HMvdW+20] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HYL+17] Po-Yao Huang, Ye Yuan, Zhenzhong Lan, Lu Jiang, and Alexander G Hauptmann. Video representation learning and latent concept mining for large-scale multi-label video classification. *arXiv preprint arXiv:1707.01408*, 2017.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[ISBX17] Zahra Iman, Scott Sanner, Mohamed Reda Bouadjenek, and Lexing Xie. A longitudinal study of topic classification on twitter. In *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[JPD+11] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence*, 34(9):1704–1716, 2011.

[KHR+18] Alexandr A Kalinin, Gerald A Higgins, Narathip Reamaroon, Sayedmohammadreza Soroushmehr, Ari Allyn-Feuer, Ivo D Dinov, Kayvan Najarian, and Brian D Athey. Deep learning in pharmacogenomics: from gene regulation to patient stratification. *Pharmacogenomics*, 19(7):629–650, 2018.

[KJTS19] Abhay Kumar, Nishant Jain, Suraj Tripathi, and Chirag Singh. From fully supervised to zero shot settings for twitter hashtag recommendation. *arXiv preprint arXiv:1906.04914*, 2019.

[KKA+21] Junaed Younus Khan, Md Tawkat Islam Khondaker, Sadia Afroz, Gias Uddin, and Anindya Iqbal. A benchmark study of machine learning models for online fake news detection. *Machine Learning with Applications*, 4:100032, 2021.

[KLH22] Vivek Kulkarni, Kenny Leung, and Aria Haghighi. Ctm–a model for large-scale multi-view tweet topic classification. *arXiv preprint arXiv:2205.01603*, 2022.

[Kos22] Simeon Kostadinov. Understanding Encoder-Decoder Sequence to Sequence Model. `https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346`, 2022.

[KSZ14] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[KTS+14] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional

neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[KZG+17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

[LC19] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.

[LCG+19] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[LCH+18] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Proceedings of the European conference on computer vision (ECCV)*, pages 201–216, 2018.

[LHH18] Jie Liu, Zhicheng He, and Yalou Huang. Hashtag2vec: Learning hashtag representation with relational hierarchical embedding model. In *IJCAI*, pages 3456–3462, 2018.

[LLS17] Yue Li, Qinghua Li, and Jie Shan. Discover patterns and mobility of twitter usersâa study of four us college cities. *ISPRS International Journal of Geo-Information*, 6(2):42, 2017.

[LLS20] Zhiyuan Liu, Yankai Lin, and Maosong Sun. Representation learning and nlp. In *Representation Learning for Natural Language Processing*, pages 1–11. Springer, 2020.

[LLXH22] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022.

[LMB+14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco:

Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[LOG+19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[LSG+21] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

[LZZP22] Junting Lei, Wanqing Zhao, Shaobo Zhang, and Jinye Peng. Multi-branch prediction network for multi-label social image classification. In *2021 Ninth International Conference on Advanced Cloud and Big Data (CBD)*, pages 201–206. IEEE, 2022.

[Mar21] Daniel Jurafsky James H. Martin. N-gram Language Models. `https://web.stanford.edu/~jurafsky/slp3/3.pdf`, 2021.

[MBP+20] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey, 2020.

[MDP+11] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

[MXY+14] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L Yuille. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*, 2014.

[NHK17] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 299–307, 2017.

[NS16] Frank Nielsen and Ke Sun. Guaranteed bounds on the kullback-leibler divergence of univariate mixtures using piecewise log-sum-exp inequalities. *arXiv preprint arXiv:1606.05850*, 2016.

[NTN20] Hideki Nakayama, Akihiro Tamura, and Takashi Ninomiya. A visually-grounded parallel corpus with phrase-to-region linking. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4204–4210, 2020.

[NZPS16] Teng Niu, Shiai Zhu, Lei Pang, and Abdulmotaleb El Saddik. Sentiment analysis on multi-view social data. In *International Conference on Multimedia Modeling*, pages 15–27. Springer, 2016.

[PC+19] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[PCCH16] Soujanya Poria, Iti Chaturvedi, Erik Cambria, and Amir Hussain. Convolutional mkl based multimodal emotion recognition and sentiment analysis. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 439–448. IEEE, 2016.

[PD07] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007.

[pdt20] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[Per21] Christophe Pere. Data representation in NLP. `https://towardsdatascience.com/data-representation-in-nlp-cc9460f855a7`, 2021.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle,

A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[PMLP18] Hai Pham, Thomas Manzini, Paul Pu Liang, and Barnabas Poczos. Seq2seq2sentiment: Multimodal sequence to sequence models for sentiment analysis. *arXiv preprint arXiv:1807.03915*, 2018.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[Rip07] Brian D Ripley. *Pattern recognition and neural networks*. Cambridge university press, 2007.

[RMM+17] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7008–7024, 2017.

[RNS+18] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[RP19] Chaudhary Jashubhai Rameshbhai and Joy Paulose. Opinion mining on newspaper headlines using svm and nlp. *International journal of electrical and computer engineering (IJECE)*, 9(3):2152–2163, 2019.

[RWC+19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[SCN19] Arka Sadhu, Kan Chen, and Ram Nevatia. Zero-shot grounding of objects from natural language queries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4694–4703, 2019.

[SDCW19] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[SGS15] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[Sti07] Stephen M Stigler. The epic story of maximum likelihood. *Statistical Science*, pages 598–620, 2007.

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[SVS+14] Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández. Syntactic n-grams as machine learning features for natural language processing. *Expert Systems with Applications*, 41(3):853–860, 2014.

[Sze90] Richard Szeliski. Fast surface interpolation using hierarchical basis functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):513–528, 1990.

[TCHB12] Pang-Ning Tan, Sanjay Chawla, Chin Kuan Ho, and James Bailey. *Advances in Knowledge Discovery and Data Mining, Part II: 16th Pacific-Asia Conference, PAKDD 2012, Kuala Lumpur, Malaysia, May 29-June 1, 2012, Proceedings, Part II*, volume 7302. Springer, 2012.

[TL19] Quoc-Tuan Truong and Hady W Lauw. Vistanet: Visual aspect attention network for multimodal sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 305–312, 2019.

[V+99] Ellen M Voorhees et al. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82, 1999.

[VdMH08] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[VKFU15] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. *arXiv preprint arXiv:1511.06361*, 2015.

[VRD+15] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[VWZL19] Sunny Verma, Chen Wang, Liming Zhu, and Wei Liu. Deepcu: Integrating both common and unique latent information for multimodal sentiment analysis. In *International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2019.

[WLHL18] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):394–407, 2018.

[WWA+19] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. Npa: neural news recommendation with personalized attention. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2576–2584, 2019.

[WWC+19] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4581–4591, 2019.

[WYW+] K Wang, Q Yin, W Wang, S Wu, and L Wang. A comprehensive survey on cross-modal retrieval (2016). *arXiv preprint arXiv:1607.06215*.

[WYW+16] Kaiye Wang, Qiyue Yin, Wei Wang, Shu Wu, and Liang Wang. A comprehensive survey on cross-modal retrieval. *arXiv preprint arXiv:1607.06215*, 2016.

[XMYR16] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[XWD+19] Xinyu Xiao, Lingfeng Wang, Kun Ding, Shiming Xiang, and Chunhong Pan. Deep hierarchical encoder–decoder network for image captioning. *IEEE Transactions on Multimedia*, 21(11):2942–2956, 2019.

[YCJL16] Quanzeng You, Liangliang Cao, Hailin Jin, and Jiebo Luo. Robust visual-textual sentiment analysis: When attention meets tree-structured recursive neural networks. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 1008–1017, 2016.

[YDY+19] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[YKSG14] Shuang-Hong Yang, Alek Kolcz, Andy Schlaikjer, and Pankaj Gupta. Large-scale high-precision topic modeling on twitter. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1907–1916, 2014.

[YLHH14] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

[YLJY15] Quanzeng You, Jiebo Luo, Hailin Jin, and Jianchao Yang. Joint visual-textual sentiment analysis with deep neural networks. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1071–1074, 2015.

[YLJY16] Quanzeng You, Jiebo Luo, Hailin Jin, and Jianchao Yang. Cross-modality consistent regression for joint visual-textual sentiment analysis of social multimedia. In *Proceedings of the Ninth ACM international conference on Web search and data mining*, pages 13–22, 2016.

[YS19] Ikuya Yamada and Hiroyuki Shindo. Neural attentive bag-of-entities model for text classification. *arXiv preprint arXiv:1909.01259*, 2019.

[YW10] Shaozhi Ye and S Felix Wu. Measuring message propagation and social influence on twitter. com. In *International conference on social informatics*, pages 216–231. Springer, 2010.

[YYS+21] Hui Yin, Shuiqiao Yang, Xiangyu Song, Wei Liu, and Jianxin Li. Deep fusion of multimodal features for social media retweet time prediction. *World Wide Web*, 24(4):1027–1044, 2021.

[ZLY21] Sun Zhang, Bo Li, and Chunyong Yin. Cross-modal sentiment sensing with visual-augmented representation and diverse decision fusion. *Sensors*, 22(1):74, 2021.

[ZZS+19] Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Li Guo. Dan: Deep attention neural network for news recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5973–5980, 2019.