UNIVERSIDAD DE GUANAJUATO

CAMPUS IRAPUATO - SALAMANCA
DIVISIÓN DE INGENIERÍAS

"Multimodal Model Based on Image and Text for Predicting Users' Interests on the Pinterest Social Network"

A thesis presented for the degree of:
Maestría en Ingeniería Eléctrica
(Instrumentación y Sistemas Digitales)

By:

Ing. Areli Cabrera Oros

Thesis Directors:

Dr. Juan Carlos Gómez Carranza
M. I. Jonathán de Jesús Estrella Ramírez

Salamanca, Guanajuato **February 2025**

UNIVERSIDAD DE GUANAJUATO

CAMPUS IRAPUATO - SALAMANCA
DIVISIÓN DE INGENIERÍAS

"Modelo Multimodal Basado en Imágenes y Texto para Predecir Intereses de Usuario en la Red Social Pinterest"

Para obtener el grado de:
Maestría en Ingeniería Eléctrica
(Instrumentación y Sistemas Digitales)

Presenta:

Ing. Areli Cabrera Oros

Directores de Tesis:

Dr. Juan Carlos Gómez Carranza
M. I. Jonathán de Jesús Estrella Ramírez

Salamanca, Guanajuato **Febrero 2025**

Abstract

The increased use of social media has led to a substantial growth in user-generated information in recent years. Revealing user interests within these platforms represents an opportunity to generate recommendation systems, conduct market research, and take advantage of this massive amount of information. This thesis proposes developing a multimodal model based on images and text to predict user interests within the social network, Pinterest. In this model, text and images are transformed through Word-embeddings and Deep-learning models to optimize a logistic regression classifier for each model and modality independently. The construction of the Mix-Modality model is performed through the combination of six different models. Of these, four are based on selecting the logistic regression models with the best score (two for images, two for text) and the remaining two on fine-tuned models of BERT and RoBETa used exclusively as text classifiers. The combination of the models based on late fusion is generated through a weighted sum according to the effectiveness of each model for predicting user interests. In addition, a comparison with two other fusion methods discussed in the literature is presented, where the fusion is made by applying a lambda factor that affects images directly or using a feature vector crossing technique. Considering the top-k accuracy metric, the results show the Mix-Modality model consistent with better results than the unimodal models and the two fusion methods.

Resumen

El incremento del uso de las redes sociales ha llevado a un crecimiento sustancial de información generada por los usuarios en los últimos años. Revelar los intereses de los usuarios dentro de estas plataformas, representa una oportunidad para generar sistemas de recomendaciones, realizar estudios de mercado, y aprovechar esta enorme cantidad de información. Este trabajo de tesis, propone el desarrollo de un modelo multimodal basado en imágenes y texto para predecir intereses de usuario dentro de la red social Pinterest. En el cual, el texto e imágenes son transformados a través de modelos de Word-embeddings y Deep-learning respectivamente, para luego optimizar un clasificador de regresión logística para cada modelo y modalidad independiente. La construcción del modelo Mix-Modality es realizada a través de la combinación de seis modelos diferentes. De los cuales, cuatro, se basan en la selección de los modelos LR con mejor puntuación, (dos para imágenes, dos para texto) y los dos restantes en modelos fine-tuned de Transformers BERT y RoBETa usados exclusivamente como clasificadores de texto. La combinación de los modelos basados en fusión tardía se genera a través de una suma ponderada de acuerdo a la efectividad de cada modelo para la predicción de los intereses de usuario. Además, se presenta una comparación con otros dos métodos de fusión abordados en la literatura, donde la fusión se realiza aplicando un factor lambda que afecta a las imágenes directamente o utilizando una técnica de cruce de vectores de características. Los resultados, considerando la métrica de top-k accuracy, muestran que el modelo Mix-Modality es consistente con mejores resultados que los modelos unimodales y los dos métodos de fusión.

Dedication

To my family, especially my mother, Lilia Oros Villafaña, and my father, Pablo Cabrera Rodriguez, for all the love and support they have given me throughout my life. Their unwavering encouragement, sacrifices, and belief in my abilities have been the foundation of my personal and academic growth. Their guidance and unconditional love have shaped me into the person I am today, and I will always be grateful for their presence and influence in my journey.

Acknowledgements

Thank you to my classmates, friends, and professors who accompanied me throughout my studies, offering their support, encouragement, and invaluable knowledge every step of the way.

Special thanks to my family, who motivated me to continue my academic training, always believing in my potential and pushing me to achieve my goals.

Institutional Acknowledgements

Thanks to the University of Guanajuato and CONACYT.

Special thanks to my thesis advisors, Dr. Juan Carlos Gómez Carranza and M. I. Jonathán de Jesús Estrella Ramírez, for their invaluable support and guidance throughout this process, whose knowledge and dedication were essential to the completion of this work.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The continuous growth of social media has become of great importance in these times, influenced by the accessibility of mobile devices, internet connectivity, and user interaction [9]. These platforms range from socializing to job seeking and interacting with companies [10]. The global impact of social networks nowadays is essential, as they are a source of valuable and helpful information for many applications, such as generating publicity, decision-making, personalized content, and more. Sharing images and posts, creating videos, and sending audio or comments on social networks have become increasingly popular for expressing thoughts and emotions, emerging as a means of communication. Therefore, the information can be presented in various modalities, such as text, images, videos, and audio.

In that sense, a multimodal approach can be beneficial for analyzing content on social networks due to the range of information formats in which people express their ideas. Also, generally on social networks, certain kind of content is related to other formats; for example, images are commonly associated with text, which aids in providing extra information about the image, such as descriptions, keywords, or tags [11]. This relationship between different information formats or modalities increases social networks as a field of study of multimodal learning. However, when incorporating several modalities, the complexity of the model increases; therefore, it is important to leverage all the information that incorporates each feature of the model. In this sequence, the fusion methods help establish a relationship among the different modalities. Additionally, recent observations suggest that representations that fuse multiple modalities yield better results than unimodal ones across various applications, including interest prediction [12].

In summary, social media represents a complex landscape, and adequate information extraction provides a valuable opportunity to tackle various tasks such as trend analysis [13], geolocalization [14], sentiment analysis [15], optimizing marketing strategies [16], and influencing in the user experience [17].

Providing a personalized and improved user experience that reflects an understanding of users' interests can lead to various outcomes on these platforms, including audience retention, increased engagement, and user satisfaction. Some applications for inferring users' interests are aimed at improving social media services, including suggesting content adapted to users, connecting users with relevant friends, encouraging interaction between them, and promoting content generation on these platforms. Personalized recommendations can also be focused on the advertising sector, an important income source for social networks [18]. The approach of generating recommendations based on users' interests is especially relevant for visual platforms where interaction between user and content can reveal valuable patterns.

As a social network, Pinterest provides a visual browser organized by interest categories. Users can save web content in image form, referred to as "pin", each pin is associated with a textual description [19]. Users can save a pin in a board, which is assigned to one of those categories. Furthermore, considering that this social network hosts 75 billion pins and boasts over 250 million monthly active users [20], the organizational question becomes relevant, as the only way to determine user preferences is by examining both features, descriptions, and categories [21]. This makes Pinterest an excellent setting for studying multimodal learning and users' interests.

This work focuses on utilizing a multimodal representation of data, integrating both images and text to infer users' interests on the Pinterest social network. The proposed approach focuses on feature extraction from different modalities using deep learning models. For transforming image data, three versions of ConvNeXt and three versions of EfficientNetV2 were proposed. On the other hand, for transforming text data, GloVe, Word2Vec, and fastText were selected. The data transformation is followed by a logistic regression machine learning model applied as a classifier to predict a category corresponding to users' interests. This approach also incorporates the exploration of using transformers as text classifiers and the application of late fusion where weights are learned from an exponential function by combining the six models that provide the best understanding of user interests, that is to say, the most accurate ones. This combination is carried out by means of a weighted sum of weights according to the accuracy of each model, making an optimization process to determine the weights that demonstrate a better performance in recognizing users' interests. The effectiveness of the proposed model, called the Mix-Modality (MM) model, is tested by comparison with two previous works that also apply fusion [22].

The hypothesis of this work is is to verify the improvement in the performance of machine learning classifiers by implementing feature extraction from both text and image modalities, followed by hyperparameter optimization of logistic regression models. This approach aims to achieve improved accuracy in a validated setting through rigorous experimentation and cross-validation techniques.

The main contributions are as follows. The construction of a mix-modalities model based on images and text applied for obtained users' interest predictions in Pinterest, where the application of late fusion is made through a weighted sum of modalities according to the relevance of each modality for the prediction of user interests. Also, a comparative analysis of fine-tuned transformers as text classifiers with the logistic regression machine learning method working with feature extraction from word-embeddings and the fusion of both modalities. The results show how the proposed method MM outperforms the models in which a single modality is used, as the comparison with the other two fusion methods in which the proposed approach achieves better and more consistent results.

1.1 Motivation

According to [23], more than 5 million people used social media in 2024, and the number is projected to be more than 6 million in 2028. In [24], it is mentioned that social media marketing strategies have proven to be highly effective for brands, thanks to the flexibility in terms of personalization and adaptation of content to the user, as well as the interaction they offer with the customer. 47% users of the Pinterest social network get on this platform to shop and discover new products. In [25] it is also mentioned that in houses whose inhabitants use Pinterest they spend 29% more than those who do not have Pinterest. These facts make Pinterest a valuable source of information for marketers and e-commerce, providing a fundamental guideline for understanding users' interests within the platform. As mentioned above, improving user experience by suggesting relevant content is also essential for social networks; in this order, to exploit the information that both the modalities, images and text of Pinterest offer, represents a basic understanding of the user's interest in the platform. In general, recognizing user interests and improving recommendation algorithms to align with visual and textual preferences is essential for social networks.

On the other hand, researchers have been studying the multimodal approach to understanding social networks, the interaction between users and the variety of information modalities, and how these interactions can reveal user behaviors or preferences. Continuous evolution in multimodal social networks context has promoted the creation of datasets [26], the development of systems for recognizing depression [27], conducting sentiment analysis [28], detecting cyberbullying [29], and generating recommendations [30]. Recent advances in multimodal recommender systems mention that a multimodal model is capable of discovering hidden relations and representing these relations between the distinct modalities, recovering information that can be forgone in unimodal approaches [31]. This observation, as well as the continued emergence of methods to analyze different modalities within social networks, raises the need to predict user interests through a multimodal approach, suggesting

that this field should be further explored and its possibilities for improvement.

The proposed approach represents an opportunity to improve the user experience on Pinterest by correctly categorizing pins corresponding to users' interests. It is projected that proper categorization is fundamental in Pinterest's taxonomy for bringing personalized recommendations to users, increasing their participation in this platform. Furthermore, the analysis of fusion between different modalities through the implemented strategies provide valuable insights into the importance of each modality for the pins' categorization, how each modality can be analyzed through a variety of models, and how each one of these models can provide different information to finally decide which model offers the most effective classification and to what extent it can contribute to the final model and the final prediction of user interests.

1.2 Objectives

The main objective of this work is to develop a multimodal learning model, called the MM model, based on text and images capable of predicting users' interest in the social network Pinterest. The model will be constructed by integrating different deep learning methods applied to each modality of text or image. It will predict users' interests based on the categories of pins uploaded by users.

Specific objectives include:

- Training the developed model by implementing feature extraction from both modalities, followed by hyperparameter optimization from the logistic regression models, and applying cross-validation techniques.
- Obtaining the two models for both modalities, text and image models, which have a better performance in classification pins from the validation subset. Based on higher accuracy from probability vectors, including transformers.
- Calculate the weights that capture the optimal classification behavior of the models. To get the best optimization for the mixed models by a ponderated sum of weights based on the models' performance.
- Test the obtained model and make a comparison with other techniques for the multimodal models in social media.

1.3 Literature Review

Recently, with the continuous rise of social networks, data that could be extracted from these platforms has led to a significant increase due to the variety of information formats users interact with, such as text, image, audio, etc. Various researchers have explored different information modalities such as image and text to tackle a

variety of tasks, like sentiment analysis [32], detecting depression or mental illness [33], fake news detection [34], extracting attributes for e-commerce [35], generating recommender systems [36].

Various researchers have addressed the task of improving user experience [37]. In the beginning, investigations were most focused on users' engagement; for example, in [38], they use a triangulation strategy of biological measurements, such as neurological and psychological (represented by body signals), combined with subjective measures, like surveys, interviews, etc., reporting the interaction between the different models to find an interaction. The objective was to see the posts on Twitter that promote users' interactions and what kind of responses they generate to monitor user engagement and experience. Nowadays, the problem is most oriented toward developed recommendation systems that focus on behavior and friendships from users in social networks [39]. Existing different compilations and analyses of works concentrate on creating recommendations tailored to users' interests [40], [39], [31]. The diversity of data on these platforms has allowed leveraging all these features to gain deeper insights into users' behavior and preferences.

In particular, recognizing user interests through a multimodal approach represents an opportunity of great interest for researchers, marketers, and even platforms. One of the first approaches was from [41] where the application of different modalities was used to create a multimodal media agent that combines text, images, videos, and audio through a series of natural language processing (NLP), deep-learning, and text-to-speech techniques to enrich the model and make intelligent and personalized suggestions to users. Their findings were tested primarily on Twitter, but they also incorporated platforms like SoundCloud, Instagram, and YouTube. In [42], video, audio, and text data are combined to build a multimodal graph based on item-item similarities, using the historical interaction item-user and fusing with the multimodal data. Their final model was tested on TikTok and Movielens, showing a better performance than other advanced multimodal systems in "learning the users' deeper preferences". In [22], the users' preference is obtained by fusing text and images and applying early and late fusion. Then, they map the fusioned features of each post. Finally, the centroid of the maps from all the posts is considered to be the users' preference. Their multimodal fusion feature approach, tested in a real-world dataset from Instagram, achieves more effective results than conventional methods.

Most of these works are adapted to a multimodal environment where fusion is a fundamental concept. This step includes how different types of information relate to each other and is an area of ongoing study. In the past, some researchers have been focusing on studying fusion effects at different stages in the process. An example of early and late-fusion performances in recognizing user preferences is represented by [22]. In their work, they propose both approaches for detecting user preferences in posts on Instagram. For early fusion, also known as feature-level fusion, they use a fully connected neural network to make the concatenation of images and text feature vectors. Then, with a VGG16 model, they extracted the combined

representation of both modalities obtained the feature map and the category for each post. For making the late fusion, also known as score or decision fusion, they first make the feature extraction through the use of TextCNN and VGG16 models, obtaining the feature map and category for text and images in a separate way, and then make the combination of features maps by using a fully connected layer and the inner product of both features maps. This work greatly represents the different stages in which the combination of modalities can be performed. However, other researchers have also been studying the utilization of cross-modality fusion. In [43], they propose using a cross-modality fusion between images and text features to detect the relationships of Twitter posts for name entity recognition. The cross-modality fusion was performed by generating nodes for each textual and image representation and creating a graph that computes the affinity between the different nodes. They calculate four types of relationships: text-text, image-image, text-image, and image-text, and integrate features from the most relevant node by a multi-head attention mask, achieving consistent and efficient results. The works mentioned above provide an understanding of how fusion works at different stages within the field of social networks. However, many others have investigated the concept of multimodal fusion to solve various problems [44], so it remains an area of great interest with room for improvement.

Both in the field of multimodal study and in that of social networks, Pinterest was chosen as a subject of study in the past. For instance, in [45], they addressed the challenge of recognizing users' interests by leveraging data from Pinterest and fusing textual and image representations. Text representations were derived from term frequencies and word embedding techniques, while image representations were obtained from a bag of visual words, pre-trained or fine-tuned CNNs. In this work, a support vector machine trains each representation individually, obtaining the final classification by combining both outputs. Similarly, in [21], employed a multimodal Boltzmann machine as a fusion method and demonstrated its efficacy in tasks such as recommending pins to boards and boards to users. Across their experiments, the multimodal approach consistently outperformed using a single modality for all recommendation tasks. In [46], the authors applied a cascade fusion method to identify Pinterest users by training each modality individually and predicting specific pin postings using a dot product. In [12], the authors demonstrated significant improvement of multimodal models under dispersed text conditions for tasks such as board recommendations and pins classification. They concentrated on handling diverse and noisy multimodal data, proposing a method in which a combination of image, words, and graphs is used to learn the semantic relationships. Also, their approach includes a method for minimizing the distance between image features and semantic relationships through the training of a deep-vision model, which learns the representations incorporating visual information and graph-based semantic relationships.

In summary, numerous methods have been implemented to extract relevant information from images and text on social media, allowing for better prediction of user interests. However, this problem remains open to experimentation with recent models to improve classification performance significantly. This thesis proposes the development of a MM model for predicting users' interests in the Pinterest social network through the late fusion of images and text. The MM model applies a series of deep learning methods to analyze image and text data and uses a logistic regression classifier to provide predictions that align with users' interests. The technique incorporates fine-tuned Transformers as text classifiers and compares the proposed approach with two previous methods.

Chapter 2

Theoretical Framework

This chapter introduces the reader to the basic concepts relevant to the research. First, the discussion of data pre-processing methods applied to image and text data is presented. Next, the discussion delves into machine learning and deep learning models, examining those considered and their operational processes. To continue exploring multimodal models emphasizing the significance of fusing different data representations. Finally, the evaluation setup and the evaluation metrics used in this work are discussed.

2.1 Pre-processing Data Methods

Social media data can be presented in various forms, such as images, text, audio signals, and videos. Obtaining the most effective representation for each data type is crucial for achieving optimal results in classification or prediction problems. As mentioned before, this work aims to jointly leverage different representations for images and text. This section will delineate the methods used to obtain these representations.

2.1.1 Text Analysis

In the realm of social media, text is the primary source of valuable information. Every post, comment, and hashtag contains textual content. NLP is employed to extract meaningful patterns to make the most of this wealth of information. However, it is common to find missing or incomplete information due to the unstructured or semi-structured nature of the language used by individuals to express their thoughts. Therefore, a critical initial step involves text pre-processing for accurate analysis. Text pre-processing comprises two key methods: feature extraction and feature selection [47].

- In *feature selection* (FS), each document represents a space model where each word (or keyword) represents a dimension. The primary aim is to delete the

most irrelevant and redundant information from the target text and assign scores to the words. Each word is assigned a score equivalent to the significance of the word in the document.

- In *feature extraction* (FE), exist two categories: morphological analysis or syntactic analysis. Morphological analysis focuses on individual words represented in a document and normally consists of tokenization and removal of stop-words as part of the pre-processing. This process will be explained further in Section 3.2. On the other hand, syntactic analysis focuses more on providing knowledge of the grammatical context for each word in a sentence.

In the past, FS was commonly used for text representation. However, this work utilizes FE via word embeddings. Word embeddings leverage contextual information from text using neural networks to assign each word a vector representation of real numbers based on its syntactic context, creating new low-dimensional vectors. This work will explore various methods of word embeddings used in literature.

GloVe

Global Vectors, usually known as GloVe, is an unsupervised learning method developed to obtain word representation based on a word-word co-occurrence matrix X. The entries in X_{ij} record the number of times the word j appears in the nearest context to the word I. Equation 2.1 represents the probability that the word j appears in the context of the word i.

$$P_{ij} = P(i|j) \tag{2.1}$$

Table 2.1 shows an example of the correlation between words *"ice"* and *"steam"* with other words trained in a 6 billion token corpus, according to [1]. In this case, we can observe that *"ice"* has a higher probability of being related to words such as *"solid"* and *"water"*, whereas *"steam"* with the word *"gas"*. When the comparison between the ratio of both probabilities is made, the higher ratio between probabilities (values much greater than 1) correlate more with *"ice"*, while the lower ratio between probabilities (values nearest to 0) correlate more with *"steam"*. In this way, we can identify which words are more closely related and which don't have a meaningful connection.

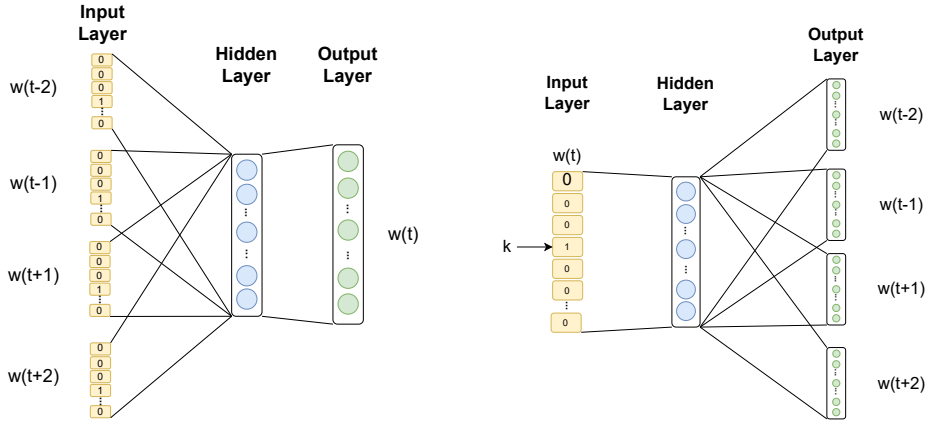
Table 2.1: Co-occurrence probabilities for *"ice"* and *"steam"* adapted from [1].

Probability and Ratio	k = *solid*	k = *gas*	k = *water*	k = *fashion*
P(k | *ice*)	1.9 x 10e-4	6.6 x 10e-5	3.0 x 10e-3	1.7 x 10e-5
P(k | *steam*)	2.2 x 10e-5	7.8 x 10e-4	2.2 x 10e-3	1.8 x 10e-5
P(k | *ice*)/P(k | *steam*)	8.9	8.5 x 10e-2	1.36	0.96

Word2Vec

Word2vec is a method for obtaining word embeddings, achieved through training two neural network models. The first model is the *"Continuous Bags of Words"* (CBOW), which predicts a middle word based on the surrounding words. The second model is the *"Continuous Skip-gram"*, which predicts the surrounding context words given a current word.

Both models take a one-hot-encoded vector as input, with all values set to 0 except for the position of the represented word (k), which is set to 1. This vector is passed through an embedding layer to create a dense, continuous vector representation. The output layer adjusts the weights to obtain a single representation (in the case of skip-gram) or the surrounding words (in the case of CBOW). The architectures of the CBOW and Skip-gram models are represented in Figure 2.1.



(a) CBOW (b) Skip-gram

Figure 2.1: Architectures of neural network models for word2vec.

FastText

FastText is a model developed for text classification and word embedding generation. Similar to word2vec, its function is based on the same two principal architectures, CBOW and Skip-grams, and it can predict an objective word or its surrounding context. The main difference between fastText and word2vec is that fastText decomposes a word in an n-gram series to assign a random vector to these n-grams, which is then adjusted for error minimization. In Figure 2.2 an example of this is shown. This approach is helpful for analyzing rare words, as it allows finding the similarities between the n-grams in place of a complete word.

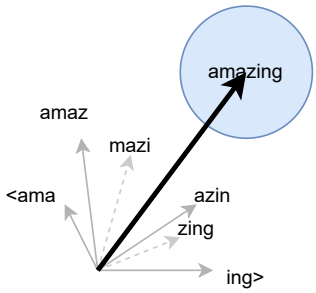


Figure 2.2: Representation of decomposition of *"amazing"* word in n-grams. with $n = 4$. Adapted from [2].

2.1.2 Image Analysis

The interpretation of images and visual information has been studied in many fields, from medicine [48] to social media [49]. Images are among the most common ways to express ideas on social networks. This work applies image feature extraction using six different current deep-learning models for image analysis. Three are based on ConvNeXt, and three are based on EfficientNetV2. Some relevant aspects of these deep-learning models are discussed below.

ConvNext

ConvNeXt is a convolutional neural network (CNN) adapted to be more competitive with transformer architectures, such as visual transformers. As discussed in [3], some of the principal attributes besides their efficiency due to their inspiration in transformers, are their simplicity and the macro design changes, where it is remarkable to note that the computational resources used by ConvNeXt are less than those of swing-transformers, which is their comparative model.

Table 2.2: Parameters of selected variants for ConvNeXt models.

Model	Channels	Blocks	Parameters
ConvNeXtTiny	96, 192, 384, 768	3, 3, 9, 3	28.6M
ConvNeXtSmall	96, 192, 384, 768	3, 3, 27, 3	50.2M
ConvNeXtBase	128, 256, 512, 1024	3, 3, 27, 3	88.5M

Table 2.2 shows the parameters of the versions used in this work, which are different only in the number of channels and blocks; also, Figure 2.3 shows the basic

architecture of the network for the tiny version.

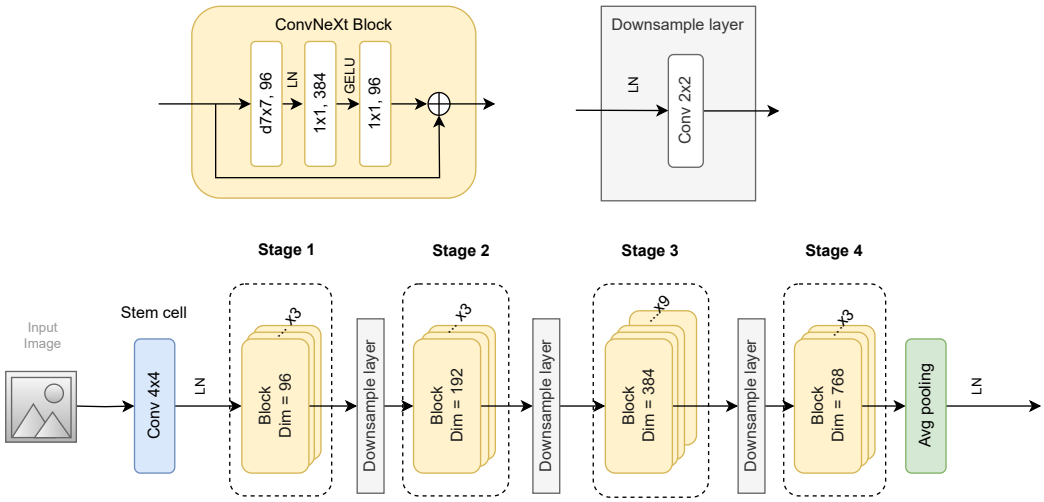


Figure 2.3: ConvNeXtTiny architecture according to [3], where LN and GELU correspond to the normalization layer and Gaussian error linear unit, respectively.

EfficientNetV2

EfficientNetV2 is a group of neural networks developed in 2021 for image recognition. Their main objective is to optimize their predecessor, EfficientNet, improving the performance of this task. As discussed in [4], some essential aspects of EfficientNetV2 are the optimization of the network design process and the balancing of MBConv blocks with fused-MBConv blocks.

Figure 2.4 shows the configuration of the EfficientNetV2 network, composed of seven stages with MBConv and fused-MBConv blocks distributed alternatively. In the blocks, SE represents a squeeze-excitation module that compresses spatial information into a single dimension and then excites the relevant channels.

Some differences between EfficientNetV2 and EfficientNet are the expansion ratio and number of layers. In EfficientNetV2, a smaller ratio decreases memory access. Also, EfficientNetV2 adds more layers to compensate for the reduced kernel size 3x3. The result of all these modifications is an optimized network group that outperforms the previous version. Table 2.3 shows the selected versions of EfficientNetV2 applicable to this work, as well as the number of channels, blocks, and parameters.

2.2 Classifiers

Classification is one of the principal tasks of supervised learning in Machine Learning. According to [50]: *" a classifier is any function that will assign a class label to an object x"*. It could be defined by Equation 2.2

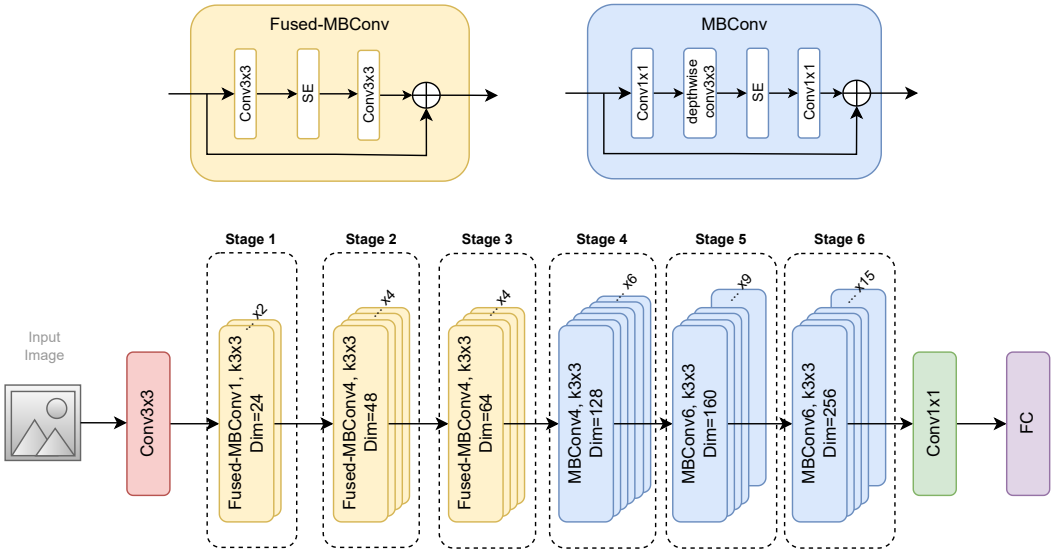


Figure 2.4: EfficientNetV2-S architecture according to [4].

Table 2.3: Parameters of selected variants for EfficientNetV2 models.

Model	Channels	Blocks	Parameters
EfficientNetV2-B1	16, 32 ,48 ,96 ,112, 192	2, 3, 3, 4, 6, 9	8.2M
EfficientNetV2-B3	16, 40, 56, 112, 136, 232	2, 3, 3, 5, 7, 12	14.5M
EfficientNetV2-S	24, 48, 64, 128, 160, 256	2, 4, 4, 6, 9, 15	21.6M

$$f_c : \mathbf{R}^n \to \Omega \quad , \quad f_c(x) = \omega \quad , \quad x \in \mathbf{R}^n \quad , \quad \omega \in \Omega \tag{2.2}$$

where R^n represents the feature space, and the object x is the feature vector which contains n features. Ω corresponds to the set of possible labels ω for this object. So, f_c is our classifier function, assigning a label to each x.

In the machine learning context, classifiers could be divided into generative and discriminative:

- *Generative.* They model the set of features for each class using $P = (x, c)$ as the joint probability of features x and class label c, which means learning the distinct features for each class, allowing the generation of new data based on the previous.
- *Discriminative.* They model $P = (c|x)$, representing the probability of class labels given the features. They focus on defining the boundary and specific differences that separate the classes.

This work applies a Logistic Regression (LR) classifier to predict users' interests, which corresponds to a discriminative classifier. Additionally, some classifiers based on the transformer's architecture are also considered for modeling textual data.

2.2.1 Logistic Regression

LR is a discriminative classifier commonly used in many tasks, including NLP. It decides on a class based on an input observation from a test set. This decision is based on vectors of weights and bias terms previously learned with a training set. The weights w_i indicate how important a feature x_i is for a class, and it could be positive or negative. Equation 2.3 illustrates the process of making a decision. To compute z, each weight is multiplied by its respective feature, resulting in a comprehensive sum of all weighted features. Then, the bias term b is added, resulting in the weighted sum of z for the class, as is shown in Equation 2.3.

$$z = \left(\sum_{i=1}^{n} w_i x_i \right) + b = \mathbf{w} \cdot \mathbf{x} + b \tag{2.3}$$

However, note that z does not necessarily correspond to a probability since their range could be $-\infty$ to ∞. z is a number to create a probability in the *sigmoid function* showed in Equation 2.4, also known as the *logistic function.* That is the reason for the method's name. The range for a sigmoid function corresponds to (0, 1), as we can note in Figure 2.5. That makes it perfect for being a probability function.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.4}$$

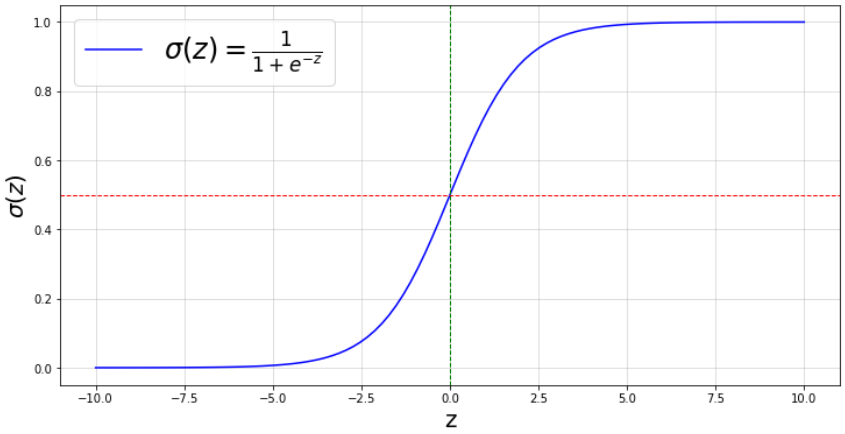


Figure 2.5: Sigmoid function.

To properly make the sigmoid function a probability, it is crucial to ensure that the sum for all classes equals 1. For example, if we had two possible classes, y = 1 and y = 0. That is $P(y = 1) = 1 - P(y = 0)$. So

$$P(y = 1) = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$$

$$P(y = 0) = 1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b)$$

For the property of sigmoid function: $1 - \sigma(x) = \sigma(-x)$

$$P(y = 0) = \sigma(-(\mathbf{w} \cdot \mathbf{x} + b))$$

The decision is made by Equation 2.5:

$$\text{decision}(x) = \begin{cases} 1 & \text{if } P(y = 1 \mid x) > 0.5 \\ 0 & \text{in other case} \end{cases} \tag{2.5}$$

The above equations are used when we have only two possible classes; it could be true or false, possible or negative, etc. For multiclass applications (e.g., positive, negative, and neutral), multinomial logistic regression, also known as softmax regression in Equation 2.6, is applied. The algorithm produces a vector of the length of the class number K, where each value $\hat{y}$ represents the estimated probability for each class k. The multinomial logistic classifier takes a generalized version of the sigmoid function to compute the probability $p = (y_k = 1 \mid x)$.

$$softmax(\mathbf{z}_i) = \frac{\exp(\mathbf{z}_i)}{\sum_{j=1}^{K} \exp(\mathbf{z}_j)} \quad 1 \leq i \leq K \tag{2.6}$$

Developing the equation 2.6, it is turned into Equation 2.7:

$$softmax(\mathbf{z}_i) = \left[\frac{\exp(\mathbf{z}_1)}{\sum_{j=1}^{K} \exp(\mathbf{z}_i)}, \frac{\exp(\mathbf{z}_2)}{\sum_{j=1}^{K} \exp(\mathbf{z}_i)}, ..., \frac{\exp(\mathbf{z}_K)}{\sum_{j=1}^{K} \exp(\mathbf{z}_i)} \right] \tag{2.7}$$

Note that the denominator $\sum_{j=1}^{K} \exp(\mathbf{z}_i)$ normalizes the probabilities.

Furthermore, the logistic regression algorithm needs to apply two concepts for the coefficients of the weights and the bias term. The first one is the distance between the actual label ($\hat{y}$) and the goal label (y); the name of this distance is the *loss function*. The second concept is the algorithm that will be used to update the weights, minimizing the loss function simultaneously; the most widely used is the *gradient descent*.

Loss Function

For the loss function, the goal is to use a function that maximizes the probability of the correct output $p = (y \mid x)$ and, at the same time, minimizes the likelihood of the incorrect. For two possible correct solutions (0 and 1), Equation 2.8 is used, where if the system output y is equal to 1, $p(y \mid x)$ is reducing to $\hat{y}$, and when $y = 0$ is reducing to $1 - \hat{y}$.

$$p(y \mid x) = \hat{y}^y (1 - \hat{y})^{1-y} \tag{2.8}$$

The log can be taken on both sides of the equation, which will also maximize the log of probability, and the sign must be changed to obtain a loss function. The result is the cross-entropy loss L_{CE} in Equation 2.9.

$$L_{CE}(\hat{y}, y) = -logp(y \mid x) = -[ylog\hat{y} + (1 - y)log(1 - \hat{y})] \tag{2.9}$$

Finally the definition $\hat{y} = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$ is carried on, resulting in Equation 2.10:

$$L_{CE}(\hat{y}, y) = -logp(y \mid x) = -[ylog\sigma(\mathbf{w} \cdot \mathbf{x} + b) + (1 - y)log(1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b))] \tag{2.10}$$

For the multinomial logistic regression, the probability $p = (y \mid x)$ is changed by $p = (y_k = 1 \mid x)$ resulting in Equation 2.11

$$L_{CE}(\hat{y}, y) = -log \frac{exp(\mathbf{w_c} \cdot \mathbf{x} + b_c)}{\sum_{j=1}^{K} exp(\mathbf{w_j} \cdot \mathbf{x} + b_j)} \tag{2.11}$$

where c is the positive class.

Gradient Descent

As mentioned above, it is necessary to minimize the loss function and obtain the optimal weight for this task. This is the goal of the gradient descent algorithm.

In Equation 2.12, note that L_{CE} is parameterized by the weights, represented by $\theta = w, b$ in the logistic regression case.

$$\hat{\theta} = argmin \frac{1}{m} \sum_{i=1}^{m} L_{CE}(f(x^{(i)}; \theta), y^{(i)}) \tag{2.12}$$

This algorithm finds the minimum in a function by calculating the gradient of the loss function, leaving a random initialization point and moving in the opposite direction, as is shown in Figure 2.6 and calculated by Equation 2.13. The step size of moving in the gradient descent algorithm is the slope magnitude in the loss function $\frac{d}{dw} L(f(x; w), y)$ weighted by a parameter called learning rate η:

$$w^{t+1} = w^t - \eta \frac{d}{dw} L(f(x; w), y) \tag{2.13}$$

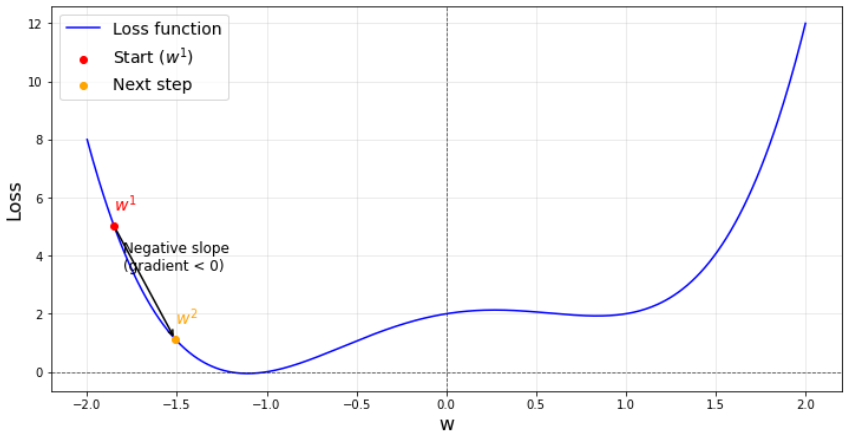


Figure 2.6: Gradient descent algorithm.

In Figure 2.7, the visualization of the gradient in a two-dimensional space is represented by a red arrow. The w parameter represents a point in a space of a high dimensionality, not only two dimensions. For each feature x_i, a weight w_i exists. The goal is to determine the influence of a small chance in w_i representing the function L. For each dimension w_i, the slope is represented by the loss function partial derivative $\frac{\partial}{\partial w_i}$. In Equation 2.14 ∇ represents the gradient, and $f(x; \theta)$ is representing $\hat{y}$.

$$\nabla L(f(x; \boldsymbol{\theta}), y) = \begin{bmatrix} \frac{\partial}{\partial w_1} L(f(x; \boldsymbol{\theta}), y) \\ \frac{\partial}{\partial w_2} L(f(x; \boldsymbol{\theta}), y) \\ \vdots \\ \frac{\partial}{\partial w_n} L(f(x; \boldsymbol{\theta}), y) \\ \frac{\partial}{\partial b} L(f(x; \boldsymbol{\theta}), y) \end{bmatrix} \tag{2.14}$$

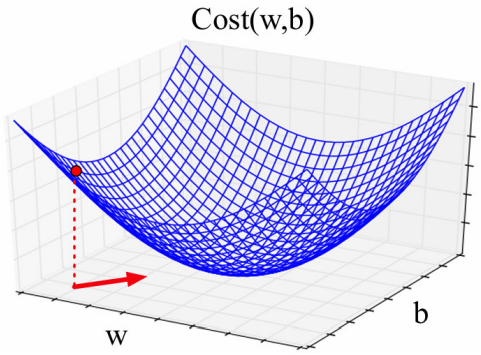


Figure 2.7: Gradient vector like a red arrow in a two-dimensional space. Taken from [5].

In the application of the logistic regression algorithm, the partial derivative of Equation 2.10 and the equivalent form is shown in Equation 2.15

$$\frac{\partial L_{CE}(\hat{y}, y)}{\partial \mathbf{w}_j} = -(y - \hat{y})\mathbf{x}_j \tag{2.15}$$

It is remarkable that the gradient with respect to w_i is equal to the difference between the system output and the real output multiplied by the respective feature x_i.

2.2.2 Transformers For Text Classification

Transformers represent a neural network architecture that primarily relies on attention mechanisms rather than recursion. Self-attention has emerged as the pioneering method in transduction [51]. Notable implementations of this architecture, such as BERT and RoBERTa, have been developed for NLP tasks, including text classification. Additionally, these architectures emphasize the encoder module of Transformers shown in Figure 2.8, utilizing a multi-layer, bi-directional approach.

BERT

BERT is a model developed for Google and widely used in NLP tasks. It is named for Bi-directional Encoder Representations from Transformers. The most important concept surrounding BERT is bi-directionality, which generates comprehension in both senses of a word, which means their surrounding context. Combined with the functionality of self-attention in Transformers, the model can correctly ponder the weights for each word and capture the differences in the context. For example, differentiating a verb from a noun in a sentence. Furthermore, embedding training was used in a 30,000-word vocabulary corpus.

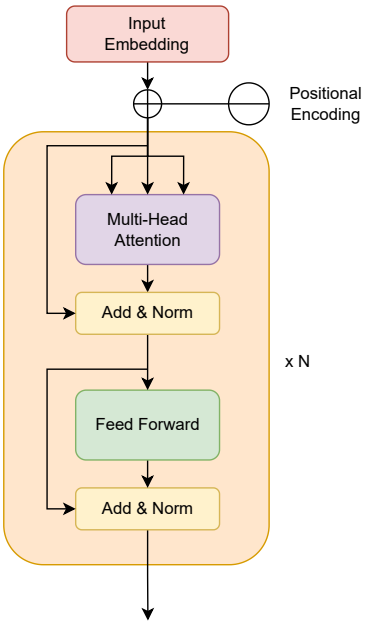


Figure 2.8: Transformer encoding module. Adapted from [6].

The diagram from BERT architecture is shown in Figure 2.9; note that encoder modules, internally, correspond to the transformers encoding modules in Figure 2.8.

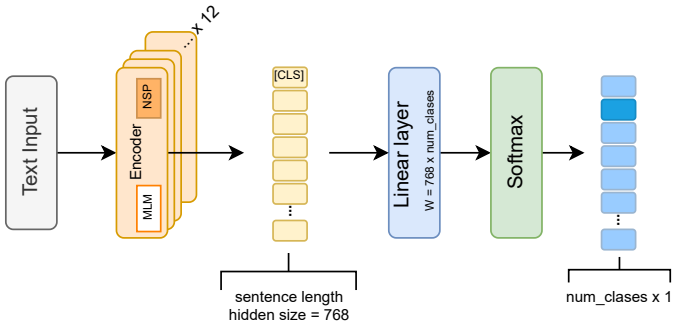


Figure 2.9: BERT model architecture according to [7].

For the text input module, the first token has an initial special token for classification [CLS]; the rest of the sentences are separated by special tokens [SEP] for making sentence differentiation, as represented in Figure 2.10. The input embedding to the encoders is the sum of the token, segment, and position embeddings.

The "masked language model" (MLM) achieves bidirectionality. The mask of one token in a sentence aims to predict the token, taking into account only the surrounding context. In the pre-training phase, the next sentence prediction (NSP)

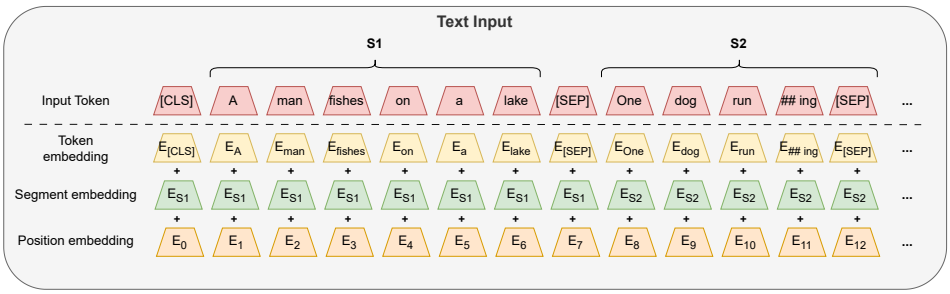


Figure 2.10: Text input module in BERT with tokenization of a sentence.

approach enriches the context when there are more than two sentences.

Finally, to make the classification, as is notable at the end of the diagram in Figure 2.9, the model uses a linear block and a softmax block to compute the probabilities for each class and then select the maximum argument. Mathematically, the process is very similar to logistic regression, differentiating the entry, which includes the feature vector obtained from the encoder of transformers and the CLS, as well as SEP tokens from BERT. Futhermore, BERT has two variants; their respective parameters are shown in Table 2.4.

Table 2.4: Parameters of variants for BERT models where BERT base (*) is the selected model to work in this research. To facilitate the reading of the thesis, in the following chapters, the use of BERT-Base will be replaced only for BERT.

Model	Channels	Blocks	Parameters
BERT-Base*	768	12	110 M
BERT-Large	1024	16	340 M

RoBERTa

The robustly Optimized BERT Approach (RoBERTa) is a model developed by Facebook AI. Principal improvements in [8] were the augmentation of data for training; while BERT is training in a dataset of 16 GB, RoBERTa was trained with a larger dataset of 160 GB, increasing time, in addition, a larger batch size and deleted the prediction of the next sentence in the pre-training phase. Other fundamental changes are the increase in sequence length during training and the change in the Dynamic MASK on the sentences. With the above mentioned, the architecture of RoBERTa is very similar to BERT (see Figure 2.11).

Note the main differences between MLM in BERT and DMLM in RoBERTa. In DMLM, the masked word changes as often as the data set increases. RoBERTa was pre-trained with a dataset 10 times larger than BERT, which means the masking has

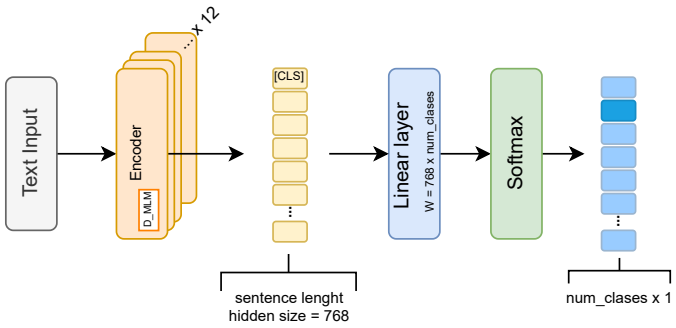


Figure 2.11: RoBERTa model architecture according to [8].

10 different ways of being presented throughout the 40 training epochs in the encoder block (see Figure 2.12). The second difference is the total number of parameters, which is 125 M for RoBERTa.

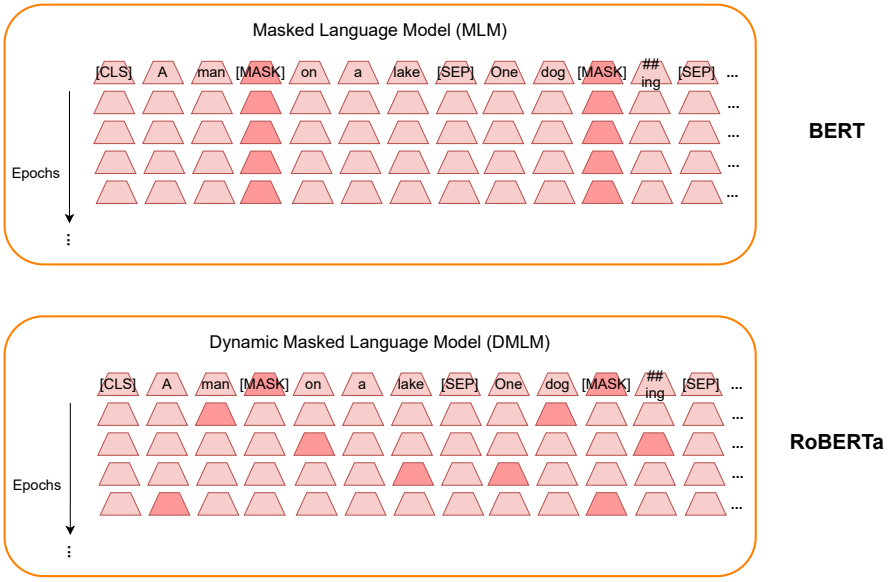


Figure 2.12: Static masking from BERT vs. RoBERTa Dynamic masking.

2.3 Feature Fusion in Multimodal Models

Multimodal modeling is an advanced approach to data analysis that integrates and combines different types of information, such as text, audio, video, and images [52]. The primary objective of a multimodal model is to enhance the accuracy and reliability of analysis systems by leveraging the unique strengths of each modality to overcome their respective limitations.

In summary, multimodal modeling represents a valuable approach in cases where the integration of diverse data types is necessary. The combination of various modalities offers a more precise representation of classes, thereby facilitating the analysis of a wide range of tasks.

Fusion of data is a form of multimodal modeling that involves combining various sets of collected data to improve decision-making by providing additional information. Researchers primarily focus on three levels of fusion: feature-level fusion, also known as early fusion; late fusion, also known as decision fusion; and cross-modality fusion [53].

- In *feature fusion or early fusion*, different types of data are transformed and fused before the classification phase. This type of fusion can provide better task fulfillment by correlating different features. On the other hand, the main features of different modalities can differ between them. Therefore, it is important to process them in the same format to detect their correlation.

- When using *decision fusion or late fusion*, features from various modalities are classified independently and then combined to achieve the final classification. This fusion approach offers the benefit of categorizing similar types of data, and each modality can be classified using the most appropriate model to capture its distinct features.

- Finally, *cross-modality fusion* can occur at any point in the model, providing a deep comprehension of how multiple modalities relate at different points in the model.

2.4 Evaluation Setup

This section presents the performance metrics and techniques for evaluating machine learning and deep learning models.

2.4.1 Dataset Split

A proper evaluation of machine learning and deep learning models is done by splitting data into training and test sets. The training set is used to build the model. In contrast, the test set evaluates the learning with new samples, allowing not only the performance but even the verification of possible problems during learning, like over-fitting. Common ways to split datasets into train-test sets are 80-20 and 90-10. Furthermore, in some cases, it is useful to use a third set for previous testing; in that way, a validation set can be applied, which is a sub-division from the training set and is commonly applied in the method's optimization phase.

Another useful technique to apply in smaller datasets and for model optimization is the *k-fold cross-validation*. It consists of dividing the training dataset into k folds, taking the selected subset k as the validation set and the rest of the groups as an only training set. The selected subset k for validation is switched until all the sets are reviewed. The cross-validation allows testing of different versions of the classification method on every iteration, which is useful in optimization applications; while, in smaller datasets, it is able to assess the reliability of the model. Normally, this division takes over the original training set, and the value k is chosen by the user, being typical values 5 and 10.

2.4.2 Performance Metrics

Having a measure of the performance model is fundamental to knowing its quality. Metrics can give a numerical representation of the sample's classification, allowing us to monitor the classification model's performance. For each classification problem, the output of the model can give us the following results: True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The first two represent the correct label from a document. In contrast, the last two represent when the model incorrectly tags a document that originally had a positive or negative label, respectively. A *confusion matrix* is a visual representation to organize these results, as can be seen in Figure 2.13.

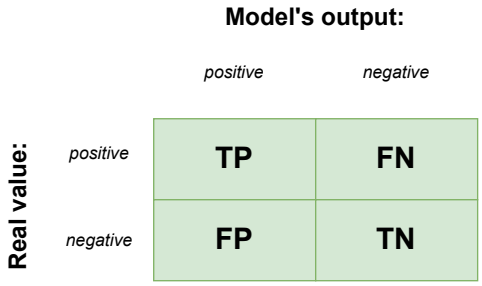


Figure 2.13: Confusion matrix for a binary classification.

With the existing information about the classification made by the model, several metrics can be used to evaluate its performance. The most commonly used metric is *accuracy* (Equation 2.16), which measures the proportion between the correct predictions or number of times the model got it right (TP, TN) and the total predictions (TP, TN, FP, FN).

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{2.16}$$

This work incorporates the use of the top-k accuracy metric, which measures how often the valid class of a sample is in the most probable k classes predicted

by the model. It is commonly used in problems where the model returns a list of probabilities instead of only the most probable class. Principally has three entries: the true labels, predicted labels, and the value of k. The metric is computed by Equation 2.17

$$\textit{Top-k accuracy} = \frac{1}{N}\sum_{i=1}^{N} 1(y_i \in Y_i^K) \tag{2.17}$$

where N is the total number of samples, y_i is the true class of i sample, Y_i^K is the conjunct of k classes with the higher probability predicted for the i sample, and $1(\cdot)$ is the indicator function which values 1 if y_i is in the k most probably classes, and 0 if not. The following example illustrates how this metric is calculated.

Imagine a problem with three samples, where $y_{true} = [C_2, C_1, C_3]$, and the predictions of the model (y_{pred}) are in Equation 2.18, and $k = 2$.

$$y_{pred} = \begin{bmatrix} C_0 & C_1 & C_2 & C_3 \\ 0.1 & 0.3 & 0.5 & 0.1 \\ 0.45 & 0.15 & 0.35 & 0.1 \\ 0.25 & 0.25 & 0.2 & 0.3 \end{bmatrix} \tag{2.18}$$

The first step is to sort the probabilities by row, resulting in the following:

- sample 1: $[C_2, C_1, C_0, C_3] \rightarrow$ Top-2: $[C_2, C_1]$. The class is correct in Top-2
- sample 2: $[C_0, C_2, C_1, C_3] \rightarrow$ Top-2: $[C_0, C_2]$. The class is wrong in Top-2
- sample 3: $[C_3, C_0, C_1, C_2] \rightarrow$ Top-2: $[C_3, C_0]$. The class is correct in Top-2

Finally, the top-k accuracy for this particular example is computed in Equation 2.19

$$\textit{Top-2 Accuracy} = \frac{1}{3}(1 + 0 + 1) = \frac{2}{3} = 66.66\% \tag{2.19}$$

Due to the model always success into the two best predictions, the top-2 accuracy is 66.66% in this case. This metric can be useful where the model can be "near" to the correct prediction without being necessary. In this work, the application of top-k accuracy metric was used for the multi-class problem through the scikit-learn module from Python [1].

[1] `https://scikit-learn.org/stable/`

Chapter 3

Methodology

This chapter discusses different phases of developing the proposed Mix-Modalities model (MM). The process begins by extracting raw data, followed by a pre-processing phase for text and images. Next, the dataset is divided into training, validation, and testing sets. This process will be followed by a systematic separation of the images and the text. For each modality, a feature extraction or transformation is performed to train and test a machine learning model, allowing us to know the importance of this particular modality to infer the users' interests.

This research explores two methods for text classification: one that uses word embeddings to build an LR classifier and another that employs a transformer-based model that works directly with the text as a classifier. Both methods can be identified in Figure 3.1 on training, validation, and testing modules, such as the word embedding and transformer-based methods blocks.

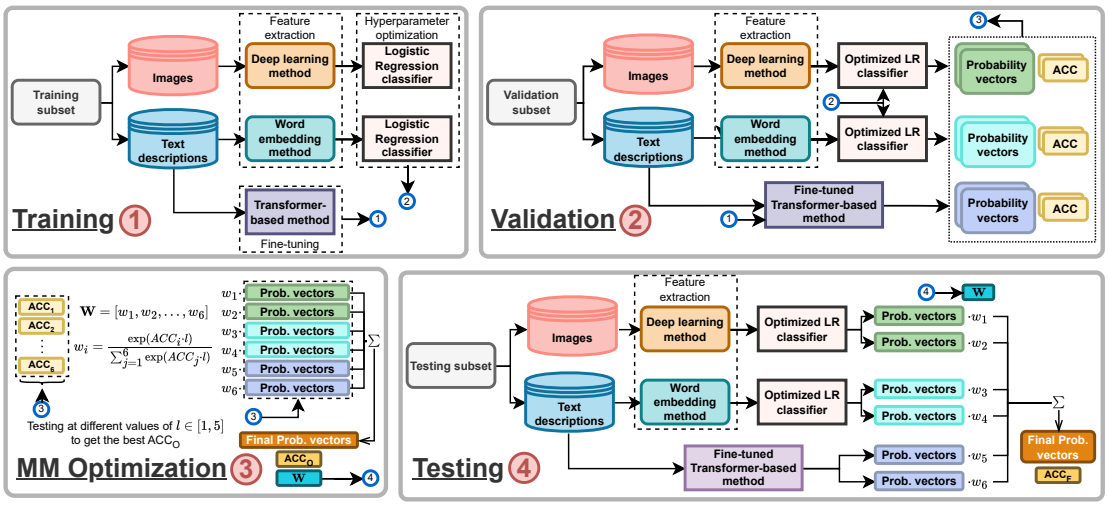


Figure 3.1: General process to obtain the MM model and the final predictions Diagram.

The first approach involves feature extraction for text and images, using word embeddings and CNN models, and for each representation, creating a classifier based on logistic regression, where its output is utilized for fusion. In contrast, the second approach employs Transformer-based models that work directly on text, producing outputs that proceed straight to the fusion step. In phase two, the validation module is used to test the optimized classifiers and obtain the first probability vectors, which will serve for the next step. Data fusion takes place during phase three (MM optimization) to get the best parameters for the model. Finally, in phase four, the final MM model is tested with a test subset (new information the model has not seen before) to generate the final predictions needed to determine users' interests.

3.1 Dataset Description

The dataset was obtained directly from the Pinterest website. It originally comprised 1,069,477 pins from 670 users. In this work, a Pin is represented as P and is composed of an image and text $P = (p_{img}, p_{txt})$, an example of a pin's composition is illustrated in Figure 3.2.

Figure 3.2: Composition of a Pin in a board and a series of boards per user.

All the pins are organized into 34 predefined categories, including categories "none" and "other." Table 3.1 details the valid categories representing specific interests, excluding the "none" and "other" categories. Additionally, there are around 400,000 pins that do not belong to any of these categories. The distribution of the raw data is presented in Figure 3.3.

The original dataset contains repeated or corrupted images, long words, short descriptions, and special characters. However, it is notable that the number of pins varies in each category (ranging from 1,440 to 77,000) and for each user (ranging from 1 to 39,000). Furthermore, since our interest lies in the pin category, it is worth noting that the dataset exhibits unbalanced classes in the number of pins per

Table 3.1: Predefined Pinterest categories.

1. Animals	12. Geek	23. Photography
2. Architecture	13. Hair beauty	24. Products
3. Art	14. Health fitness	25. Quotes
4. Cars & motorcycles	15. History	26. Science & nature
5. Celebrities	16. Holiday events	27. Sports
6. Design	17. Home decor	28. Tattoo
7. DIY crafts	18. Humor	29. Technology
8. Education	19. Illustration posters	30. Travel
9. Film music books	20. Kids	31. Weddings
10. Food and drink	21. Men fashion	32. Women fashion
11. Gardening	22. Outdoors	

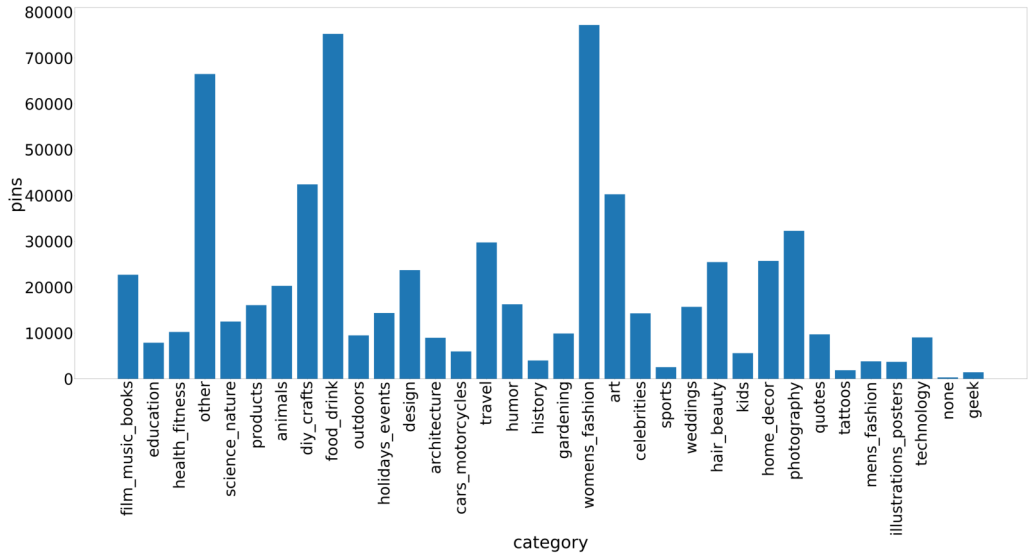


Figure 3.3: Raw data distribution.

category, as shown in Figure 3.3. To enhance the dataset's quality, a pre-processing was performed to address these issues and any other potential difficulties.

3.2 Data Pre-processing

On social media, people often express their ideas in an easy, effortless manner. As a result, the data collected from these platforms, such as posts, comments, and criticisms, can be unstructured. Since data quality greatly impacts how our model interprets data, pre-processing is crucial in achieving the best results.

The approach begins as shown in Figure 3.4 by extracting raw data, followed by a pre-processing phase that begins with text tokenization, removing of stopwords, short or very long words, and short descriptions. For the second modality, duplicated images were removed. This process will be followed by a systematic separation of the images and the text.

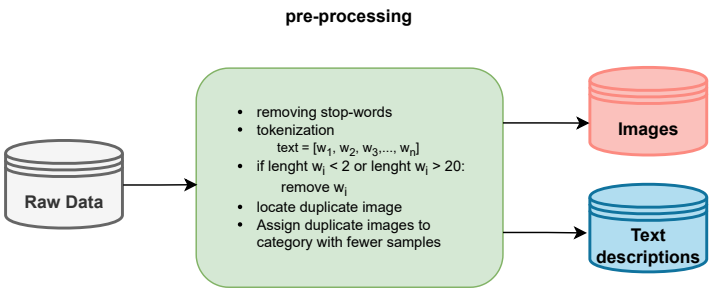


Figure 3.4: Pre-processing raw data and separation between images and text.

3.2.1 Text

First, the textual description of pins undergoes a pre-processing phase. In cases where word embeddings are used, techniques such as tokenization are employed. Tokenization involves splitting a document into individual chunks of words called tokens [54]. For example, consider the sentence: "beautiful photo from Canada." The tokens for this sentence are "beautiful," "photo," "from," and "Canada." This step is crucial for normalizing the text. Another technique applied in the model is stop word removal. Some words, such as prepositions and articles, are irrelevant for specific analyses when processing text. Removing words like "an," "as," "is," and "the" can avoid unnecessary calculations and allow a focus on more relevant words for specific purposes. For instance, in a food-related context, the word "cook" might appear frequently and be associated with greater relevance.

In our specific dataset, we removed stop-words and then made a tokenization where words with less than two characters and more than 20 were not considered.

Also, concise descriptions with fewer than four words were not considered. The NLTK [1] module in Python was employed to stop word removal and general text processing.

3.2.2 Image

On the other hand, the image module from the Python Imaging Library [2] (PIL) is employed for image preprocessing. As mentioned before, the primary focus during image preprocessing is identifying and eliminating corrupted and duplicate images through a grayscale comparison. If an image is assigned to multiple categories, the duplicate image is placed in the category with the fewest samples to achieve a balanced dataset.

3.2.3 Separation and Selection of Text and Images

The data distribution after deleting the corrupted data, pins without descriptions or descriptions with less than four words, and pins without any category or assigned in "none" and "other" categories, is visualized in Figure 3.5, giving a dataset of 171,778 pins. An example of the distribution of pins per category for a random user is shown in Figure 3.6 to compare the dataset before and after the pre-processing.

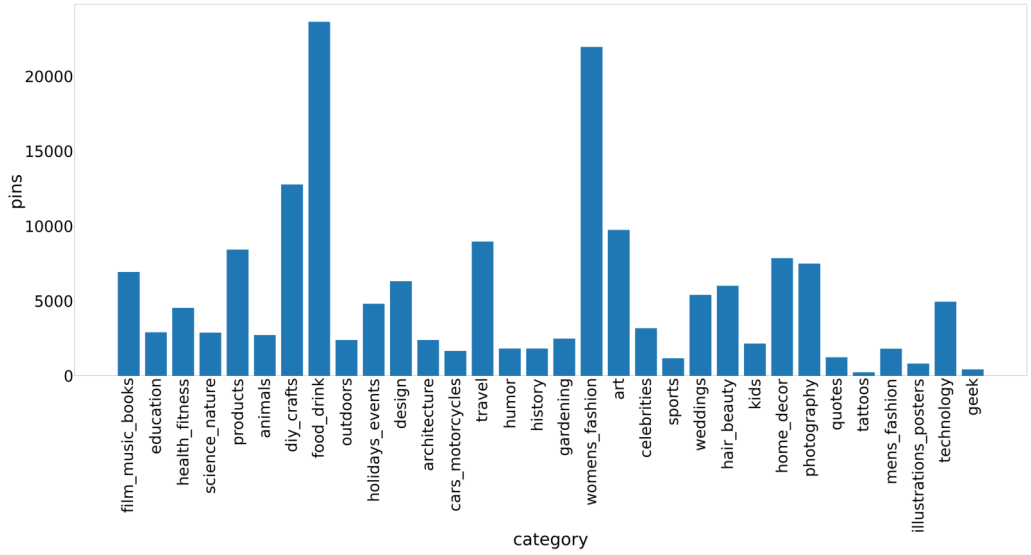


Figure 3.5: pins per category (unbalanced classes).

Upon analysis, it is noticeable in Figure 3.5 that some categories contained thousands of pins, while others had only a few hundred. Only categories with more

[1] https://www.nltk.org
[2] https://pillow.readthedocs.io/en/stable/

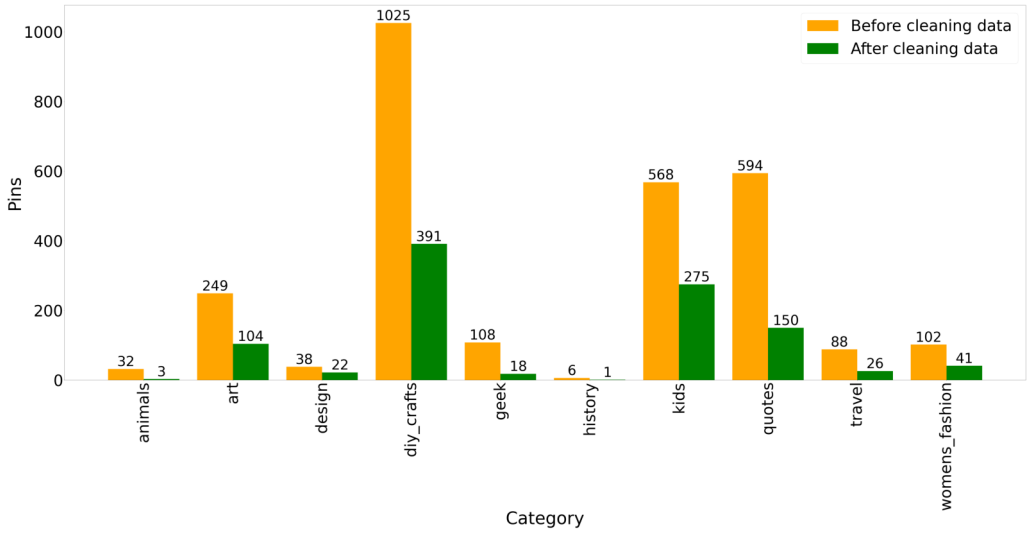


Figure 3.6: Distribution of pins for a random user before and after the pre-processing of the dataset.

than 1000 pins were included to address the issue of unbalanced categories. Consequently, the categories: *"[12] week"*, *"[19] illustration posters"*, and *"[28] tattoo"* were removed. Subsequently, for each remaining category, 1000 pins were randomly selected, resulting in a final dataset containing 29 categories, with 29000 pins in total, each pin composed of an image and textual description. The final dataset for the experiments was split into 90% for training and 10% for testing in a stratified way. Similarly, the 90% was split into 80% for training and 20% for the validation phase, where data is utilized for optimization in the fusion phase.

The dataset is segmented into the two pin's components: text and images. This categorization enables the use of word embeddings and vector image representations, leveraging the main features of each modality. Two distinct methods are utilized when dealing with text data. The first method generates word embeddings and necessitates implementing all the pre-processing steps outlined in Section 3.2.1. The second method is geared towards classifying text using transformers, where the only pre-processing step applied to the text descriptions involves removing URLs and converting all text to lowercase.

3.3 Data Transformation

The objective in this phase is to take the image and the textual description of a pin and turn it into two separate vectors for each modality that captures their main characteristics:

$$P = (v_{img}, v_{txt})$$

3.3.1 Text Transformation

For the first approach and to obtain the best word embedding representation, the cleaned text of the pin is passed as separate words: $p_{txt} = [w_1, w_2, w_3..., w_n]$, through different pre-trained text models. Each text model $\mathbf{M}_{txt}$ has a pre-trained vector for each word, as described in Equation 3.1.

$$\mathbf{u}_j = \mathbf{M}_{txt}(w_j) \tag{3.1}$$

Then, the average of each word's vectors of p_{txt} is computed to obtain a vector as follows:

$$\mathbf{v}_{txt} = \frac{1}{n}\sum_{j=1}^{n}\mathbf{u}_j \tag{3.2}$$

The chosen models for building the word embedding were trained on large corpora of news (Word2Vec), social media (GloVe), or web pages in English (fastText) and have demonstrated good performance in problems related to social media. Table 3.2 shows the dimensions of the word vectors for each model.

Table 3.2: Pre-trained word embeddings vector length.

Model	**Pre-trained on**	**Vector size**
fastText	Wikipedia	300
Word2Vec	Google news	300
GloVe	Twitter	200

In the second scenario, this process is inherently integrated into the transformer architecture. The method through which transformers generate text vectors is detailed in Section 2.2.2. The specific pre-trained data for the used transformer models are listed in Table 3.3.

Table 3.3: Pre-trained Transformer vector length.

Model	**Pre-trained on**
BERT	BooksCorpus
RoBERTa	Common Crawl News

The only adjustment to the models' configuration as classifiers was the learning rate, which was set to 8.35e−0.6. The training process took seven epochs, and a batch size of 6 was utilized to accommodate limitations in computational resources.

3.3.2 Image Transformation

In the case of the images, obtaining the feature vector v_{img} involves first resizing the image to 224 x 224 to be in the correct format for the different image models $\mathbf{M}_{img}$ (ConvNextBase, ConvNextSmall, ConvNextTiny, EfficientNetV2B1, EfficientNetV2B3, EfficientNetV2S) implemented using the Keras[3] module in Python.

All models are loaded with pre-trained weights, as shown in Table 3.4, from ImageNet [4], a dataset with over 1 million images and 1000 classes commonly used in deep learning research. The use of pre-trained models helps reduce implementation costs. The top classification layer is skipped to obtain the feature vector v_{img}, and global pooling is applied to average the features.

$$v_{img} = \mathbf{M}_{img}(p_i) \tag{3.3}$$

Table 3.4: Pre-trained vision models used for the images transformation.

Model	**Pre-trained on**	**Vector size**
ConvNeXtTiny (CNT)	ImageNet	768
ConvNeXtSmall (CNS)	ImageNet	768
ConvNeXtBase (CNB)	ImageNet	1026
EfficientNetV2B1 (ENv2B1)	ImageNet	1280
EfficientNetV2B3 (ENv2B3)	ImageNet	1536
EfficientNetV2S (ENv2S)	ImageNet	1280

3.4 General Process

3.4.1 Training

For the training phase, 90% of the data is considered, and it is divided into 80% and 20% to obtain the validation subset. In the first scenario, feature extraction is applied to both modalities using word-embeddings and image models. After extracting feature vectors from the image and text data, experiments were conducted using a machine-learning classifier. Separate LR classifiers with a maximum of 10,000 iterations were trained for each data modality, according to each extractor model. Following this, an optimization process was performed involving the adjustment of the regularization parameter, exploring a range of values such as $[0.1, 0.01, 0.001, 0.0001]$, with the optimal value of C selected to achieve the highest accuracy. Also, a 5-fold

[3]https://keras.io/api/applications/
[4]https://www.image-net.org

cross-validation was implemented in the optimization process. The resulting optimized classifier of this phase will be used in the validation phase. All experiments were executed using Python with the scikit-learn[5] module. In the diagram (Figure 3.1), this step is referred to as "Hyperparameter optimization". In contrast, the second approach processes the text through a fine-tuned transformer.

3.4.2 Validation

A validation subset was used to apply the optimized version of the LR classifier to a smaller portion of the data. For the first scenario, the final optimized LR classifiers are used in phases 2 and 4, validation and testing; each produces a probability vector containing 29 probabilities, one for each class. During the validation phase, a pin P is transformed as a vector (v_{img}, v_{txt}) to be then passed through a determined optimized classifier. In this point of the process, the output of each LR-optimized classifier is the probability vector for each model. For image modality six different probability vectors, one for each image model tested,

$$b_{img} = [b_{img_1}, b_{img_2}, b_{img_3}, b_{img_4}, b_{img_5}, b_{img_6}]$$

For the first approach, where word embeddings are taken into account, exist three probability vectors, one for each text model tested:

$$b_{txt_1} = [b_{txt_{1-1}}, b_{txt_{1-2}}, b_{txt_{1-3}}]$$

The second approach has two probability vectors as a direct output from the fine-tuning transformer process for the text modality:

$$b_{txt_2} = [b_{txt_{2-1}}, b_{txt_{2-2}}]$$

In resume, for each pin, the existing probability vectors are

$$p_i = [b_{img}, b_{txt_1}, b_{txt_2}]$$

Where sub-index in text modality indicates the different approaches, word embeddings and transformers; in total, eleven probability vectors for each pin. The main objective of this phase is to choose models with a higher probability for image models and word-embeddings, while the two from transformers were selected directly.

3.4.3 Mix-Modalities Optimization

Mixing the modalities is carried out by six fused models: two image models, two text models for word embeddings, and two models from transformers. In the cases

[5]https://scikit-learn.org/stable/

of images and word embeddings, they must be accompanied by their respective LR-optimized classifier. The weights of the $\mathbf{W} = [w_1, w_2, ..., w_6]$ vector are calculated by Equation 3.4.

$$w_i = \frac{\exp\left(acc_i \cdot l\right)}{\sum_{j=1}^{6} \exp\left(acc_j \cdot l\right)} \tag{3.4}$$

The selection criteria for the image and word embedding models is the accuracy achieved in the validation step, which will correspond to acc_i in Equation 3.4. This accuracy is multiplied by a constant parameter l. In this case, the l tested values were $[1, 2, 3, 4, 5]$, and the denominator normalizes the weights. In this sense, for each l value, a weights vector $\mathbf{W}_l = [w_1, w_2, ..., w_6]$ is obtained, and for each l value the following process is done: the weights are multiplied by each model's respective probability vectors, smoothing out any potential errors in the models' predictions and leveraging each model's strengths. Finally, these probability-weighted vectors are summed to obtain a single probability vector used to compute the accuracy. In this phase, the primary goal is to obtain a weight vector that maximizes classification performance (accuracy).

3.4.4 Testing

Given the final weight vector $\mathbf{W}_l$, corresponding to the selected l value, the final model is tested with the 10% of data. The selected six models, four with their respective LR optimized classifier, for images and word embeddings cases and the two from transformers probability vectors are multiplied by the $\mathbf{W}_l$ vector and then summed to obtain the final probability vector with Equation 3.5.

$$PP_P = \sum_{n=1}^{6} \mathbf{b}(n) * \omega_n \tag{3.5}$$

where n denotes the actual model to transform text or image data, the vector $\mathbf{b}$ represents the predicted probabilities for this representation n. A probability vector (PP) of length 29 is derived for a pin P. The distinction lies in the fact that these probabilities are based on various representations. Therefore, there is the flexibility to consider as many representations as necessary. To leverage the full potential of the models used for images and text feature extraction, the results from these different models are utilized as representations. The final classification performance of the MM model is obtained with the accuracy metric and the top-k accuracy. Figure 3.7 shows a visual representation of the fusion for an example pin.

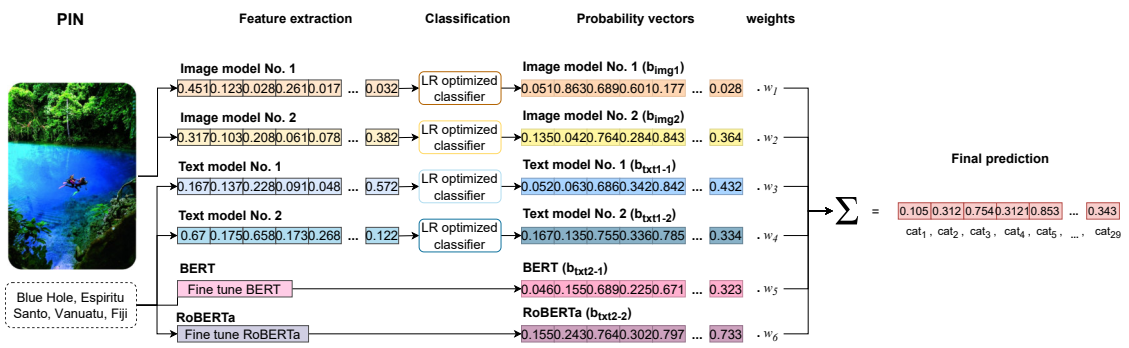


Figure 3.7: Example of fusion with six selected models.

Chapter 4

Results

In this chapter, we present the results of our experiments, organized into tables and plots. Each experiment is briefly described, followed by discussions related to their respective graphics.

The results demonstrate the performance of our methodology in the following order. First. LR optimization for single models, cases for text and images individually. Second, following results for transformers as text classifiers. Then, late fusion of LR optimized classifiers for word-embeddings and transformers. Finally, the inclusion of the two comparative methods like feature-cross and late fusion with lambda, and the proposed approach.

To facilitate the view of the plots, the abbreviation of the respective models is in Table 4.1

Table 4.1: Models abbreviations.

Model	**Abbreviation**
ConvNextBase	CNB
ConvNextSmall	CNS
ConvNextTiny	CNT
EfficientNetV2B1	EfNv2B1
EfficientNetV2B2	EfNv2B3
EfficientNetV2S	EfNv2S
fasttext	fTxt
GloVe	gV
Word2Vec	w2v
BERT	BT
RoBERTa	RbTa

4.1 Logistic Regression Single Models

4.1.1 Word Embeddings

The results for optimizing LR with text models showed that in all cases, the value for the regularization parameter was $C = 0.1$. The GloVe model had the best accuracy, with 0.4628. An explainable reason for its better performance could be its pre-training in a social network environment, as in the case study, in contrast with fast text and word2vec, which were pre-trained in Wikipedia and Google News, respectively.

Table 4.2: Results of LR applied to only text models.

Text Model	**fastText**	**GloVe**	**Word2Vec**
Accuracy	0.3686	**0.4624**	0.4331

4.1.2 Image Models

Bellow in Table 4.3, the performance for the Image models is shown. The regularization parameter in all the optimized LR classifiers was C = 0.01, except for CNB, which was $C = 0.001$. The best accuracy value is for the ConvNeXtBase model with an accuracy equal to 0.4803, representing an improvement of at least 5% concerning the other models, which can be attributed to the more significant number of parameters than the other image models.

Table 4.3: Results of LR applied to only Images models.

Image Model	**CNB**	**CNS**	**CNT**	**EfNv2B1**	**EfNv2B3**	**EfNv2S**
Accuracy	**0.4803**	0.4362	0.4251	0.4320	0.4282	0.4068

4.2 Transformers as Classifiers

The results of fine-tuning the transformers for text classification are shown in Table 4.4. We observe improved performance when compared to word embeddings. The best model is RoBERTa, with an accuracy of 0.5513

Finally, Figure 4.1 illustrates the top-k accuracy for images and text as individual models, including the transformers results for text classification.

Table 4.4: Classification performance of the fine-tuned transformers.

Text Model	**RoBERTa**	**BERT**
Accuracy	**0.5513**	0.5493

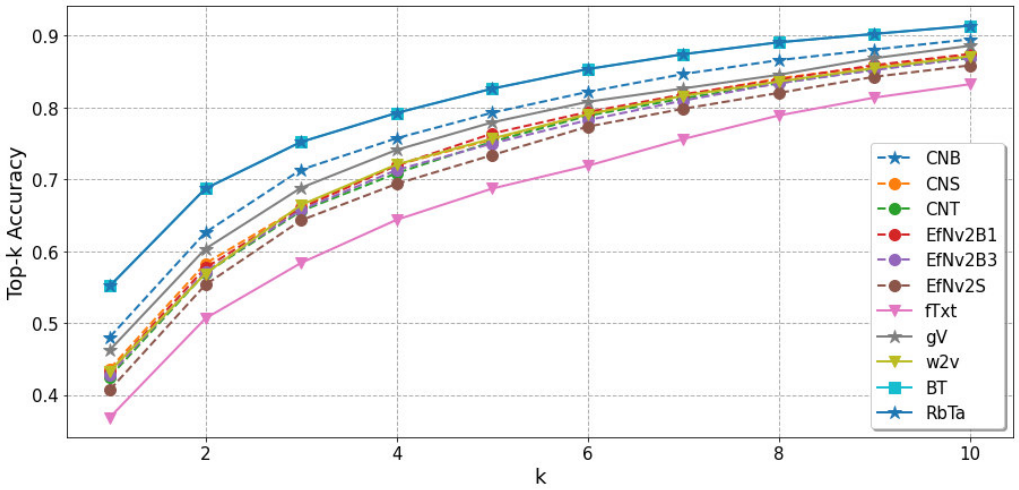


Figure 4.1: Top-k accuracy for individual models with logistic regression classification for images (◯), word embeddings (▽), and transformers as a classifiers (□). The best performance for each modality is marked with (⋆).

4.3 Logistic Regression Fusion

4.3.1 Late Fusion with Lambda

Exploring the methodology applied in [45], once the probability vectors for each pin are obtained after the LR optimized classifiers. The fusion must be performed using the weighting factor parameter λ, which enables control over the importance assigned to text or image. The following equation determines the weighted sum for the vectors:

$$\mathbf{f}_P = \lambda * \mathbf{b}_i + (1 - \lambda) * \mathbf{b}_t \tag{4.1}$$

In the context of the analysis, the vector $\mathbf{f}_P$ denotes the final probabilities for a pin $\boldsymbol{P}$. Various values for the parameter λ are being explored in the experiments, specifically, the values $[0, 0.3, 0.5, 0.7, 1]$. When these values are applied to equation 4.1, the results can be interpreted as follows: when $\lambda = 0$, the text is prioritized, and images are disregarded; when $\lambda = 1$, images are prioritized, and text is disregarded; and when $\lambda = 0.5$, equal importance is assigned to both text and images.

Results for the application of late fusion with equation 4.1 and being similar

to our first method in where word-embeddings and images models are applied, are shown in Figure 4.2, and resumed in Table 4.5. Notably, a significant increase exists compared to models where only a single modality was considered. The combination of models that gives the best accuracy value is the fusion between the ConvNeXtBase model and GloVe, having an accuracy of 0.5472 when $\lambda = 0.5$, which means giving the same importance to the text and image model. Representing an improvement of 6% in contrast with the ConvNeXtBase image model and 8% respect the GloVe model.

Table 4.5: Accuracy method 1. Where late fusion is applied with lambda factor between images and word-embedding models.

Image Model	**CNB**	**CNS**	**CNT**	**EfNv2B1**	**EfNv2B3**	**EfNv2S**
λ	**Image Models + GloVe model**					
0.3	0.5368	0.5296	0.5227	0.5289	0.5258	0.5131
0.5	**0.5472**	0.5165	0.5148	0.5248	0.5151	0.5120
0.7	0.5297	0.4855	0.4834	0.4875	0.4844	0.4768
λ	**Image Models + fastText model**					
0.3	0.5168	0.4782	0.4779	0.4741	0.4693	0.4600
0.5	0.5058	0.4648	0.4589	0.4548	0.4513	0.4427
0.7	0.4931	0.4489	0.4410	0.4410	0.4386	0.4231
λ	**Image Models + Word2Vec model**					
0.3	0.5268	0.5141	0.5048	0.5131	0.5106	0.5027
0.5	0.5279	0.4941	0.4862	0.492	0.4831	0.4775
0.7	0.5086	0.4696	0.4620	0.4658	0.4596	0.4503

The top 5 multimodal models fused with λ along with the models without any type of fusion are shown in Figure 4.2, where there is a significant improvement of over 10% when the value of k increases from 1 to 2, and similar improvements are observed for the subsequent values of k. The plot presents the top-k accuracy for the first ten values of k. The results of the best single image and word embedding models (without fusion) are represented by a dotted line marked with squares, while the top 5 fusions with the best accuracy are illustrated by a solid line with circles; finally, the fusion of the best performing models is marked with ($\star$). It is worth noting that the value of λ yielding the best fusion results is $\lambda = 0.5$.

4.3.2 Late Fusion with Lambda and Transformers

For the second approach, the Equation 4.1 is used, but apply transformers as classifiers. The results are shown in Table 4.6 using the accuracy metric.

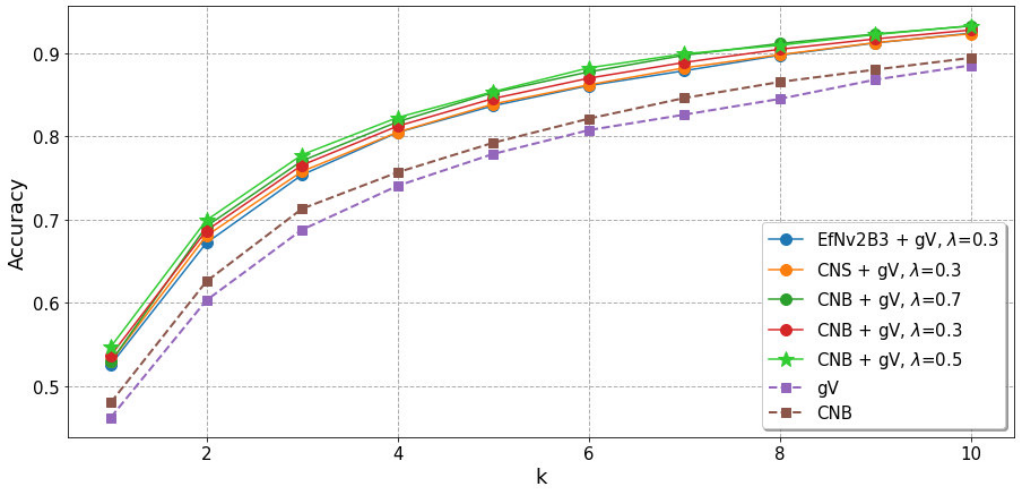


Figure 4.2: Top 5 multimodal λ fusion models compared to the best unimodal models, where the best fusion is marked with ($\star$).

Table 4.6: Accuracy method 2. Where late fusion is applied with lambda factor between images and fine-tuned transformers models.

Image Model	**CNB**	**CNS**	**CNT**	**EfNv2B1**	**EfNv2B3**	**EfNv2S**
λ	**Image Models + RoBERTa**					
0.3	0.5672	0.5675	0.5672	0.5644	0.5641	0.5637
0.5	0.5920	0.5810	0.5820	0.5851	0.5837	0.5751
0.7	0.5868	0.5606	0.5606	0.5562	0.5634	0.5617
λ	**Image Models + BERT**					
0.3	0.5724	0.5689	0.5706	0.5665	0.5648	0.5620
0.5	**0.5931**	0.5879	0.5868	0.5817	0.5813	0.5813
0.7	0.5913	0.5606	0.5644	0.5579	0.5579	0.5575

From the table, it is notable that most of the top-5 values are from a fusion between BERT and the ConvNext images models, which suggests a better performance from BERT in fusion, in contrast with the best accuracy of single models, in where RoBERTa has a higher performance.

As the same in the previous section, Figure 4.3 compares the top-k accuracy between the best fusion and the best single modalities.

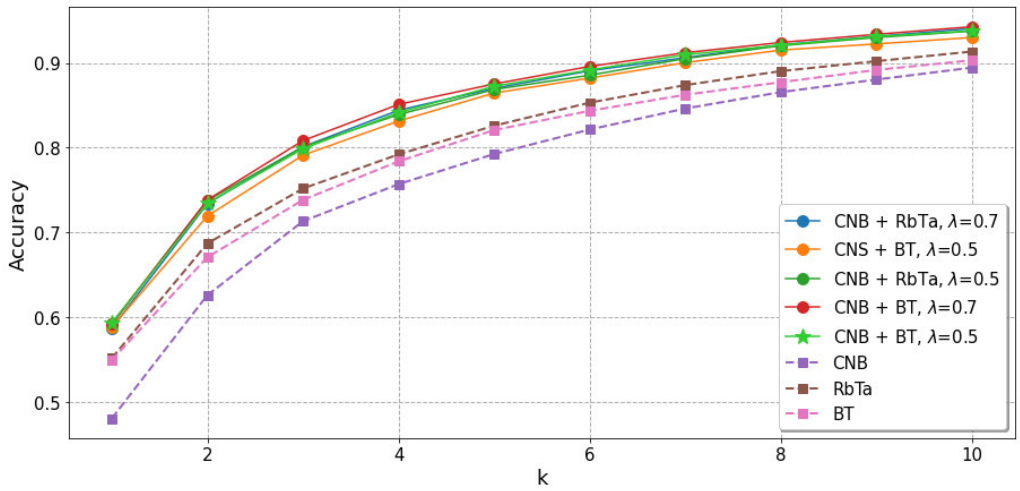


Figure 4.3: Top 5 multimodal λ fusion models compared to the best unimodal image model and both transformers, where the best fusion is marked with ($\star$).

In the plot, the higher performance of RoBERTa in an individual way is notable; furthermore, even if the accuracy of the fusion between BERT and the ConvNext Image models is higher, the top-k accuracy is increased for the fusion between RoBERTa and the models of the image when k is rising, suggesting that RoBERTa provides more consistent performance overall. For this reason, the fusion between RoBERTa and ConvNeXtBase with $\lambda = 0.5$ is taken as the comparative fusion method for this experiment.

4.3.3 Features Cross

This experiment compares against a state-of-the-art fusion method addressed in [55] by integrating the feature vectors extracted from each modality and applying an optimized LR for classification. This process involved performing a direct product ($\otimes$) between the different feature vectors, precisely the image feature vector ($\mathbf{v}_i$) and the text feature vector ($\mathbf{v}_t$). Producing a matrix $\mathbf{G}$ that effectively represents both modalities. To extract the most significant features, column-wise and row-wise max-pooling were applied to the matrix. This technique identifies the most dominant features corresponding to the text and the image data separately. Once the representative features were isolated, the resulting feature vectors were concatenated.

This final vector encompasses the most significant characteristics derived from the fusion of both models.

The results of this experiment, reflecting the performance of the fusion method, are presented in Table 4.7. The different image models are at the top of the table, and the three main divisions represent the results for each text model combination. The optimization between ConvNext and GloVe models was a regularization value $C = 0.001$, whereas, for GloVe and EfficientNetV2, $C = 0.01$. For fastText optimization, all models report a value of $C = 0.1$. For the fusion with Word2Vec, all the cases reported a value $C = 0.1$, except the combination with ConvNeXtBase, which reported a $C = 0.01$.

$$\mathbf{G}_i t = \mathbf{v}_i \otimes \mathbf{v}_i \tag{4.2}$$

Table 4.7: Feature-cross applying logistic regression.

Image Model	CNB	CNS	CNT	EfNv2B1	EfNv2B3	EfNv2S
GloVe	0.4672	0.4682	0.4617	0.4803	0.4717	0.4672
fastText	0.442	0.4537	0.4327	0.4775	0.4606	0.4444
Word2Vec	0.4882	0.4686	0.4562	**0.5179**	0.5075	0.4865

The combination that produces the best performance is between the EfficientNetV2B1 and Word2Vec models. This fusion provides an accuracy of 0.5179. Representing an improvement of 8% respects the respective models' performance as individuals. However, the result is lower than transformers as a classifier, which has an accuracy of 0.5513 and 0.5493 for RoBERTa and BERT, respectively. The fusion between image models and transformers through this method is not applicable because the fusion combines the modalities of the feature vectors. In this work, the textual description vectorization is an internal process of the transformer's models.

4.4 Mix-modalities Late Fusion

Our MM model is applied in the final experiment using Equation 3.5. A weighted sum of all the probabilities generated by the different models is calculated, with ω_n representing the weighted factor adjusted based on the relevance of the model. This approach assigns more weight to models with higher accuracy.

On the other hand, and as is mentioned in Section 3.4.3, the proposed method implements an optimization in the calculus of $\mathbf{W}_l$, by varing the l parameter used in Equation 4.3, the results of the optimization are shown in table 4.8

Table 4.8: Optimization of **W** for selection of l value.

l	**Accuracy**
1	0.6425
2	0.6421
3	0.6404
4	0.6394
5	0.6388

$$\omega_n = \frac{e^{acc_n * \ell}}{\sum_{n=1}^{6} e^{acc_n * \ell}} \tag{4.3}$$

Finally, in Table 4.9, the accuracy for each selected model is shown along with their respective weight factors calculated using Equation 4.3.

Table 4.9: Selected models for mix-modalities fusion.

Model	**Accuracy**	ω_n
CNB	0.4803	0.1703
CNS	0.4362	0.1669
RoBERTa	0.5513	0.1742
Bert-base	0.5493	0.1729
GloVe	0.4624	0.1596
Word2vec	0.4331	0.1561

The results for our model are shown in Table 4.10, where we can note the top-k accuracy for different values of k. Also, in Figure 4.4 a comparison between the performance of the other methods is plotted; where the mix-modalities late fusion increase is notable.

Table 4.10: Top-k accuracy for mix-modalities late fusion.

k	**1**	**2**	**3**	**4**
Accuracy	0.6127	0.7524	0.8200	0.8582

In general, there is a notable increase in accuracy when the proposal MM model is used, having an accuracy of 0.6127. The second better result is from method two, when the lambda method is used, but where the fine-tuned transformer is applied, with an accuracy of 0.5931. For the fusion using lambda with word embeddings

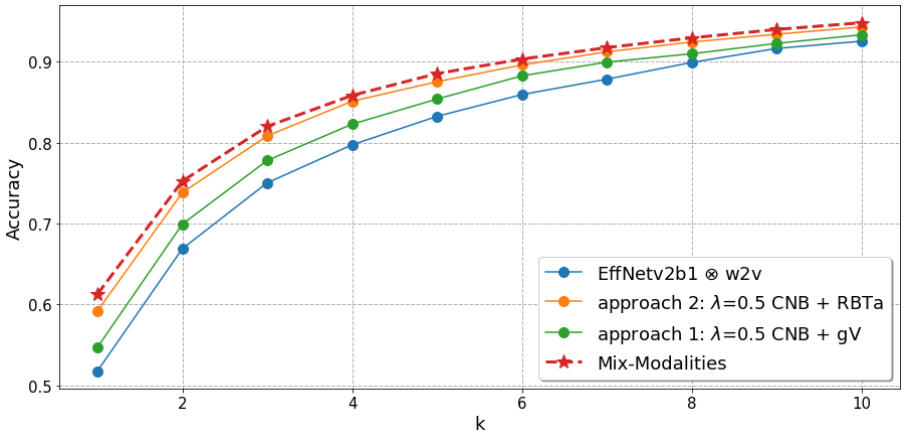


Figure 4.4: Best fusion multi-modal models vs MM model.

[45], the best result for their approach with our dataset is an accuracy of 0.5472, a combination between ConvNeXtBase and GloVe, giving the same importance to images and text ($\lambda = 0.5$). Finally, the result with a lower performance is from [55] method, with an accuracy of 0.5179 when EfNv2B1 and Word2Vec are fusion. In this case, the result is improved even for the single use of transformers as a text classifier, indicating a possible difficulty in the integration or relation between images and text features.

Chapter 5

Conclusions

This thesis presented a multimodal model based on text and images taken from the Pinterest social network, combined with late fusion, to predict users' interests on the platform. Involving the application of current deep-learning models for analyzing image data and using word-embedding models for the textual description analysis. Furthermore, individual LR classifiers for text and image were optimized to obtain a probability vector for interest. Moreover, two different fine-tuned transformers were applied as classifiers for textual descriptions, also bringing a comparison with the word-embedding approach. The study of the various methods addressed in this research allows us to leverage the multiple modalities of information and the capacity of each variety of proposed models for making an appropriate feature extraction. The proposed approach was compared with two methods previously mentioned in the state of the art, which were tested with the database mentioned in the work and the models for analyzing images and text described in this work. Based on the analysis of the results, the conclusions are as follows:

- Working with data separately provides several advantages, primarily regarding manageability and performance evaluation. By processing each modality individually, we can streamline our efforts to fine-tune models specifically designed for the unique characteristics of each data type.

- This separation facilitates easier troubleshooting and adjustment, as specific issues can be addressed without the complexity introduced by combining multiple modalities, allowing for a more precise performance comparison between single modalities, enhancing our understanding of how each contributes to effectively predicting user interests.

- The best fusion in method one, where the late fusion was applied to feature extraction through word-embeddings and images models, was between GloVe and ConvNeXtBase, when both images and text are taken equally. However, the previous result does not improve the performance of the BERT and

RoBERTa Transformers applied, suggesting that including them in the proposed approach was a good decision.

- In general, the fusion of both modalities is more effective than using a single modality, aligning with the literature of multimodal modeling and demonstrating that image and text information complement each other in this task and similar ones.

- The experiments demonstrate consistent and efficient performance of the proposed MM approach, fusing six different models compared to the alternative feature-cross and lambda methods. This illustrates how each individual's understanding of each model provides different and complementary information, even if it is the same modality.

- Results persist when the approach is extended to the use of top-k accuracy metric as the value of k increases. An improvement of more than 10% when $k = 2$ indicates how close the model is to making an accurate prediction and proposing that the suggestion of more than one category can generate a better representation of user interest.

Some limitations faced during the development of this work are: the amount of information wasted during training because it was not initially cataloged by users in the database, time and computational resources to compute the possible combinations between text and image models; the number of categories in which user interests are cataloged, representing problems such as the need for more data, difficulties in generating a feature separation boundary due to similarities between categories, or higher computational cost.

This study involves rethinking traditional approaches in the research landscape of the multimodal field, applying the understanding of various deep-learning models and approaches for each modality. Compared to previous works, we found not only how the information that provides each modality is complementary between them but also how the interpretability of each model can contribute helpful knowledge and in what measure it can be leveraged to improve the results in this particular task. Even if the actual accuracy is not relatively high, the proposed approach represents an improvement compared to the two previously tested methods, validating the proposed hypothesis. This suggests the necessity of developing more robust models.

This thesis study represents an advance in future projects and works, such as the possible integration of other fusion types, such as cross-modality or early fusion, to combine image and text and experimentation with different modalities, such as video or audio signals. Furthermore, due to continuous progress in improving or developing models for feature extraction in images and text, these can also be tested and implemented in the actual approach. Also, the effectiveness of this model can also be applied to other platforms for developing content recommendation systems.

The scope of this work outlined in the objective was achieved by developing a consistent MM model capable of integrating the models, which present a better understanding of users' interests in the Pinterest social network, demonstrating the impact of the proposed methodology.

Bibliography

[1] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[2] Ben Athiwaratkun, Andrew Gordon Wilson, and Anima Anandkumar. Probabilistic fasttext for multi-sense word embeddings. *arXiv preprint arXiv:1806.02901*, 2018.

[3] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986, June 2022.

[4] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106. PMLR, 2021.

[5] Daniel Jurafsky. Speech and language processing, 2000.

[6] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[7] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1. Minneapolis, Minnesota, 2019.

[8] Y Liu, M Ott, N Goyal, J Du, M Joshi, D Chen, O Levy, M Lewis, L Zettlemoyer, and V Stoyanov. Roberta: A robustly optimized bert pretraining approach. arxiv [preprint](2019). *arXiv preprint arXiv:1907.11692*, 1907.

[9] Alakananda Vempala and Daniel Preoţiuc-Pietro. Categorizing and inferring the relationship between the text and image of twitter posts. In *Proceedings of the 57th annual meeting of the Association for Computational Linguistics*, pages 2830–2840, 2019.

[10] Thomas Aichner, Matthias Grünfelder, Oswin Maurer, and Deni Jegeni. Twenty-five years of social media: a review of social media applications and definitions from 1994 to 2019. *Cyberpsychology, behavior, and social networking*, 24(4):215–222, 2021.

[11] Feiran Huang, Xiaoming Zhang, Jie Xu, Zhonghua Zhao, and Zhoujun Li. Multimodal learning of social image representation by exploiting social relations. *IEEE transactions on cybernetics*, 51(3):1506–1518, 2019.

[12] Lifang Wu, Dai Zhang, Meng Jian, Bowen Yang, and Haiying Liu. Multimodal joint representation for user interest analysis on content curation social networks. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 363–374. Springer, 2018.

[13] Michal Monselise, Chia-Hsuan Chang, Gustavo Ferreira, Rita Yang, and Christopher C Yang. Topics and sentiments of public concerns regarding covid-19 vaccines: Social media trend analysis. *Journal of medical Internet research*, 23(10):e30765, 2021.

[14] David Jurgens, Tyler Finethy, James McCorriston, Yi Xu, and Derek Ruths. Geolocation prediction in twitter using social networks: A critical analysis and review of current practice. In *Proceedings of the international AAAI conference on web and social media*, volume 9, pages 188–197, 2015.

[15] Sancheng Peng, Lihong Cao, Yongmei Zhou, Zhouhao Ouyang, Aimin Yang, Xinguang Li, Weijia Jia, and Shui Yu. A survey on deep learning for textual emotion analysis in social networks. *Digital Communications and Networks*, 8(5):745–762, 2022.

[16] Mazhar Javed Awan, Mohd Shafry Mohd Rahim, Haitham Nobanee, Ashna Munawar, Awais Yasin, and Azlan Mohd Zain. Social media and stock market prediction: a big data approach. *MJ Awan, M. Shafry, H. Nobanee, A. Munawar, A. Yasin et al.," Social media and stock market prediction: a big data approach," Computers, Materials & Continua*, 67(2):2569–2583, 2021.

[17] Hyun K Kim, Sung H Han, Jaehyun Park, and Wonkyu Park. How user experience changes over time: A case study of social network services. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 25(6):659–673, 2015.

[18] Fattane Zarrinkalam, Stefano Faralli, Guangyuan Piao, Ebrahim Bagheri, et al. Extracting, mining and predicting users' interests from social media. *Foundations and Trends® in Information Retrieval*, 14(5):445–617, 2020.

[19] Philipp Berger, Patrick Hennig, Daniel Dummer, Alexander Ernst, Thomas Hille, Frederik Schulze, and Christoph Meinel. Extracting image context from pinterest for image recommendation. pages 326–332, 2015.

[20] Rafael S Gonçalves, Matthew Horridge, Rui Li, Yu Liu, Mark A Musen, Csongor I Nyulas, Evelyn Obamos, Dhananjay Shrouty, and David Temple. Use of owl and semantic web technologies at pinterest. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18*, pages 418–435. Springer, 2019.

[21] Haiying Liu, Sinuo Deng, Lifang Wu, Meng Jian, Bowen Yang, and Dai Zhang. Recommendations for different tasks based on the uniform multimodal joint representation. *Applied Sciences*, 10(18):6170, 2020.

[22] Jenq-Haur Wang, Yen-Tsang Wu, and Long Wang. Predicting implicit user preferences with multimodal feature fusion for similar user recommendation in social media. *Applied Sciences*, 11(3):1064, 2021.

[23] Statista. Number of worldwide social network users, 2024. Accessed: 2025-02-20.

[24] Mohammed T Nuseir, Ghaleb A El Refae, Ahmad Aljumah, Muhammad Alshurideh, Sarah Urabi, and Barween Al Kurdi. Digital marketing strategies and the impact on customer experience: A systematic review. *The Effect of Information Technology on Business and Marketing Intelligence Systems*, pages 21–44, 2023.

[25] Andrew Lipsman. Why facebook provides scale, but instagram and pinterest offer relevance for social commerce. *eMarketer*, 2019.

[26] Dan S Nielsen and Ryan McConville. Mumin: A large-scale multilingual multimodal fact-checked misinformation social network dataset. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 3141–3153, 2022.

[27] Felipe T Giuntini, Mirela T Cazzolato, Maria de Jesus Dutra dos Reis, Andrew T Campbell, Agma JM Traina, and Jo Ueyama. A review on recognizing depression in social networks: challenges and opportunities. *Journal of Ambient Intelligence and Humanized Computing*, 11:4713–4729, 2020.

[28] Ramandeep Kaur and Sandeep Kautish. Multimodal sentiment analysis: A survey and comparison. *Research anthology on implementing sentiment analysis across multiple disciplines*, pages 1846–1870, 2022.

[29] Peiling Yi and Arkaitz Zubiaga. Session-based cyberbullying detection in social media: A survey. *Online Social Networks and Media*, 36:100250, 2023.

[30] Aghiles Salah, Quoc-Tuan Truong, and Hady W Lauw. Cornac: A comparative framework for multimodal recommender systems. *Journal of Machine Learning Research*, 21(95):1–5, 2020.

[31] Hongyu Zhou, Xin Zhou, Zhiwei Zeng, Lingzi Zhang, and Zhiqi Shen. A comprehensive survey on multimodal recommender systems: Taxonomy, evaluation, and future directions. *arXiv preprint arXiv:2302.04473*, 2023.

[32] Mohamed M Mostafa. More than words: Social networks' text mining for consumer brand sentiments. *Expert systems with applications*, 40(10):4241–4251, 2013.

[33] Sharath Chandra Guntuku, David B Yaden, Margaret L Kern, Lyle H Ungar, and Johannes C Eichstaedt. Detecting depression and mental illness on social media: an integrative review. *Current Opinion in Behavioral Sciences*, 18:43–49, 2017.

[34] M Senthil Raja and L Arun Raj. Fake news detection on social networks using machine learning techniques. *Materials Today: Proceedings*, 62:4821–4827, 2022.

[35] Tiangang Zhu, Yue Wang, Haoran Li, Youzheng Wu, Xiaodong He, and Bowen Zhou. Multimodal joint attribute prediction and value extraction for e-commerce product. *arXiv preprint arXiv:2009.07162*, 2020.

[36] Euiju Jeong, Xinzhe Li, Angela Eunyoung Kwon, Seonu Park, Qinglong Li, and Jaekyeong Kim. A multimodal recommender system using deep learning techniques combining review texts and images. *Applied Sciences*, 14(20):9206, 2024.

[37] Hsin Chen, Anastasia Papazafeiropoulou, Ta-Kang Chen, Yanqing Duan, and Hsiu-Wen Liu. Exploring the commercial value of social networks: Enhancing consumers' brand experience through facebook pages. *Journal of Enterprise Information Management*, 27(5):576–598, 2014.

[38] Jeffrey Bardzell, Shaowen Bardzell, Tyler Pace, and Jeremi Karnell. Making user engagement visible: a multimodal strategy for interactive media experience research. In *CHI'08 extended abstracts on Human factors in computing systems*, pages 3663–3668. 2008.

[39] Xiao Li, Li Sun, Mengjie Ling, and Yan Peng. A survey of graph neural network based recommendation in social networks. *Neurocomputing*, 549:126441, 2023.

[40] Viomesh Kumar Singh, Sangeeta Sabharwal, and Goldie Gabrani. Comprehensive analysis of multimodal recommender systems. In *Data Intelligence and*

Cognitive Informatics: Proceedings of ICDICI 2020, pages 887–901. Springer, 2021.

[41] M Matsiola, C Dimoulas, A Veglis, and G Kalliris. Augmenting user interaction experience through embedded multimodal media agents in social networking environments. 2005.

[42] Fei Lei, Zhongqi Cao, Yuning Yang, Yibo Ding, and Cong Zhang. Learning the user's deeper preferences for multi-modal recommendation systems. *ACM Transactions on Multimedia Computing, Communications and Applications*, 19(3s):1–18, 2023.

[43] Guoheng Huang, Qin He, Zihao Dai, Guo Zhong, Xiaochen Yuan, and Chi-Man Pun. Gdn-cmcf: A gated disentangled network with cross-modality consensus fusion for multimodal named entity recognition. *IEEE Transactions on Computational Social Systems*, 11(3):3944–3954, 2024.

[44] Arnab Barua, Mobyen Uddin Ahmed, and Shahina Begum. A systematic literature review on multimodal machine learning: Applications, challenges, gaps and future directions. *Ieee access*, 11:14804–14831, 2023.

[45] Yagmur Gizem Cinar, Susana Zoghbi, and Marie-Francine Moens. Inferring user interests on social media from text and images. In *2015 IEEE international conference on data mining workshop (ICDMW)*, pages 1342–1347. IEEE, 2015.

[46] Juan Carlos Gomez, Mario-Alberto Ibarra-Manzano, and Dora-Luz Almanza-Ojeda. User identification in pinterest through the refinement of a cascade fusion of text and images. *Advances on Language & Knowledge Engineering*, page 40, 2017.

[47] Rizwana Irfan, Christine K King, Daniel Grages, Sam Ewen, Samee U Khan, Sajjad A Madani, Joanna Kolodziej, Lizhe Wang, Dan Chen, Ammar Rayes, et al. A survey on text mining in social networks. *The Knowledge Engineering Review*, 30(2):157–170, 2015.

[48] Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. Medical image analysis using convolutional neural networks: a review. *Journal of medical systems*, 42:1–13, 2018.

[49] Sicheng Zhao, Hongxun Yao, Yue Gao, Guiguang Ding, and Tat-Seng Chua. Predicting personalized image emotion perceptions in social networks. *IEEE transactions on affective computing*, 9(4):526–540, 2016.

[50] Ludmila I Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2014.

[51] Lena Strobl, Dana Angluin, David Chiang, Jonathan Rawski, and Ashish Sabharwal. Transformers as transducers. *arXiv preprint arXiv:2404.02040*, 2024.

[52] Soujanya Poria, Erik Cambria, Rajiv Bajpai, and Amir Hussain. A review of affective computing: From unimodal analysis to multimodal fusion. *Information fusion*, 37:98–125, 2017.

[53] Ava Jackson, Rodolfo Patel, Ethan Taylor, and Mia Anderson. Fusion-enhanced multimodal social media analysis. 2024.

[54] Pansy Nandwani and Rupali Verma. A review on sentiment analysis and emotion detection from text. *Social network analysis and mining*, 11(1):81, 2021.

[55] Takumi Takahashi, Takuji Tahara, Koki Nagatani, Yasuhide Miura, Tomoki Taniguchi, and Tomoko Ohkuma. Text and image synergy with feature cross technique for gender identification. *Working Notes Papers of the CLEF*, 10, 2018.