Neurocomputing 626 (2025) 129620

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Leveraging language models for automated distribution of review notes in animated productions

Diego Garcés [a,b,*], Matilde Santos [c], David Fernández-Llorca [d,e,**]

[a] Computer Science Faculty, Complutense University of Madrid, Madrid, Spain
[b] Skydance Animation Madrid, Madrid, Spain
[c] Institute of Knowledge Technology, Complutense University of Madrid, Madrid, Spain
[d] European Commission, Joint Research Centre, Seville, Spain
[e] Computer Engineering Department, University of Alcalá, Madrid, Spain

ARTICLE INFO

Keywords:
Movie production
Review notes
Text Classification
Large Language Models (LLM)
Natural Language Processing

ABSTRACT

During the production of an animated film, professionals at the animation studio prepare thousands of notes. These notes describe improvements and corrections identified by supervisors and directors during daily meetings where the film's progress is reviewed. After each meeting, these notes are manually distributed to the appropriate departments that need to address them. Due to the manual nature of this process, many notes are not assigned correctly, and the identified issues are not addressed, reducing the final quality of the film. This article describes and compares several approaches to automatically distribute notes using multi-label text classification with different language models (LM). Implemented methods include logistic regression models, encoder-only models such as the BERT family, and decoder-only models such as Llama 2 including fine-tuning and QLoRA techniques. Training and inference were conducted on a local RTX-3090. The results of the different techniques have been compared, achieving a maximum average accuracy of 0.83 and an f1-score of 0.89 with the fine-tuned Multilingual BERT model. This demonstrates the validity of these models for multi-label text classification, as well as their usefulness in a hitherto unexplored area such as animation studios.

1. Introduction

The film production process necessitates continuous collaboration among the involved parties. Professionals develop proposals and intermediate results, which are reviewed on a daily basis. During these reviews, directors and supervisors provide comments on how the presented material can be improved or corrected. This feedback is transcribed into text notes and stored in a database using Production Tracking Software. Subsequently, the Production department manually assigns these notes to the different artistic departments responsible for addressing them. Animation production is a highly technical process. However, many tasks remain entirely manual, such as the distribution of these corrections. As a manual process, it is highly prone to errors in the fast-paced environment of producing an animated film This can lead to many notes being assigned incorrectly or omitted altogether, ultimately reducing the quality of the film.

The assignment of these notes to their respective artistic disciplines can be conceptualized as a text classification task. The notes are free-text transcriptions of verbal comments made by Directors and Supervisors during review meetings at the Animation Studio. Producers write those notes dynamically as the meeting progresses. In addition, a studio comprises various departments, as the production of an animated film demands diverse and specialized skills. Depending on the discipline in which a particular artist works, the departments may include Modelling, Surfacing, Rigging, Animation, CFX, Layout, FX, Compo, Final Layout (FL) or Lighting. In each review session, the supervisor comments on all aspects that can be improved in the material presented, regardless of the specific area the improvement pertains to. A note may address both the animation performance of a character and about the shading behaviour when deforming geometry, and in such cases, it must be handled by two different departments (in this example, Animation and Surfacing). Consequently, the assignment problem becomes a text classification task with multiple labels.

In light of this challenge, this work proposes a text note classification system that directs these notes to the appropriate departments within an animation studio. To achieve this, the material, composed of natural language text, has been pre-processed, and various language models have been adapted, implemented and tested. Three methods

* Corresponding author at: Computer Science Faculty, Complutense University of Madrid, Madrid, Spain.
** Corresponding author at: European Commission, Joint Research Centre, Seville, Spain.
E-mail addresses: digarces@ucm.es (D. Garcés), david.fernandez-llorca@ec.europa.eu (D. Fernández-Llorca).

https://doi.org/10.1016/j.neucom.2025.129620
Received 5 June 2024; Received in revised form 4 January 2025; Accepted 31 January 2025
Available online 7 February 2025

have been evaluated and compared: transfer learning with logistic regression, fine-tuning encoder-only language models, and decoder-only large language models. The results confirm the effectiveness of these techniques for this multi-label classification problem.

The main contributions of this work can be summarized as follows:

- A method to pre-process text review notes from an animation studio is proposed.
- Three methods using language models are developed and applied to the multi-label classification of these text notes.
- Several configurations of these models have been tested to enhance their performance.
- The validity of the language models for distributing text notes across various departments is demonstrated.

To the best of our knowledge, the proposed approaches for multi-label classification for the distribution of text review notes in the context of animated film production are entirely novel.

The article is structured as follows. Section 2 describes related works on the use of language models for text classification. Section 3 describes the materials, namely the database of text notes and the processing applied. Section 4 describes the three proposed classification methods, including the language models used. The results of the application of these models are presented and compared in Section 5. The article concludes with the discussion of conclusions and future work.

2. Related work

Text classification is a research area that has consistently attracted the interest of the research community for many years, due to the importance of this task and its numerous applications across different fields. General surveys such as those in [1–3], describe both traditional and modern approaches to text classification and provide comparisons between them.

Most text classification problems are single-label, where each text example or instance is associated with a single class. For example, traditional binary and multi-class classifications are sub-categories of single-label classification [4]. However, the advancement in capabilities on one hand, and the increase in the complexity of the problems addressed on the other, have led to a notable rise in the number of studies focused on classifying text instances with multiple non-exclusive labels [5]. Multi-label text classification is a generalization of the multi-class classification problem. In this sense, it is a more challenging task which requires more training data and compute to cover the entire label space. In this section, we review some of the most recent and relevant works on multi-label text classification.

First, we describe some of the most challenging variations of multi-label text classification [6]. For example, extreme multi-label text classification, where the number of labels can reach hundreds of thousands, or even millions [7]. Hierarchical multi-label text classification, where the multiple labels are constrained in a structured and hierarchical manner [8]. Imbalanced multi-label text classification, also known as long-tailed label distribution [9], which is a very common aspect of most multi-label training dataset, where there is a significant uneven distribution of samples for each label. Weakly supervised multi-label text classification, which aims to classify text into multiple categories without relying on human-labelled data [10]. And multi-label text classification with missing labels, where only a limited number of labels are annotated, and others are missing [11].

Focusing on the various fields of application, we recently encounter a wide range of topics. For instance, in [12], dialogues are categorized with up to 18 different malevolent labels, such as guilt, arrogance and violence. A very common application area is the automatic multi-topic classification of documents, which has been applied in many different domains. For example, it has been used to classify patent documents according to the Cooperative Patent Classification (CPC) system [13]. It has also been applied to classify legislative documents according to legal vocabulary such as EUROVOC [14]. Additionally, it is widely used to identify relevant keywords in scientific documents [15]. Multi-label classification and topic modelling from consumer reviews have also been widely studied in different domains, such as the fashion industry [16] and holiday rentals platforms [17]. Another domain that is naturally structured as a multi-label categorization problem is hate speech detection, which has attracted a lot of interest in the last years, with multiple contributions [18,19] and datasets [20–22]. Multi-label text classification has also been applied to the medical domain, where medical texts, such as clinical reports records [23], are classified into multiple non-exclusive medical codes [24].

Regarding the most commonly used techniques in the context of multi-label text classification, it is important to highlight that the trend has been very similar to that of text classification in general [3]. This spans from traditional methods based on feature extraction (e.g., bag-of-words, N-gram, word2vec, global vectors, etc.) followed by some form of machine learning model (e.g., Naïve Bayes, K-Nearest Neighbour, Support Vector Machine, etc.), to deep learning methods, including Convolutional Neural Networks, Recurrent Neural Networks, Graph Neural Networks, and attention-based models such as transformer-based architectures, including language models and the different types of architectures such as encoder–decoder, encoder-only or decoder-only. We direct the reader to specific surveys that provide a comprehensive analysis of the latest approaches and datasets in the context of multi-label text classification [25–27].

Similarly, it is important to highlight the recent significant growth in the use of Large Language Models (LLMs) to address multi-label text classification [28]. This can be achieved through adaptation and fine-tuning [29,30], few-shot strategies [31,32], or even via prompt engineering [25,33] and zero-shot approaches [31,32,34,35]. Since the performance of LLMs continues to grow and their usage is increasingly widespread, it is reasonable to explore these models for such problems. However, their efficiency is worse than that of smaller models fine-tuned for specific multi-label text classification tasks, and it is also not yet clear whether they can provide better results [28,36]. More studies, like the one we present in this paper, are needed to explore the problem of multi-class text classification from this perspective and across various application domains to gather more evidence.

As an overview of the analysed related works on multi-label text classification, we present Table 1, distinguishing between pre-LLMs and post-LLMs approaches. The table does not make any direct comparisons, as the application context, domains and datasets used differ in each case.

As observed in Table 1, and as far as we know, apart from our own previous works, there are no studies focused on the multi-label text classification of review notes within the field of animated film production. Review notes are usually highly technical, specific to the context of the animation studio, the film, and the reviewer. As a result, they are highly diverse, incorporating different personal styles and specific jargon. Consequently, mapping these notes to the relevant departments in a non-exclusive manner poses a significant challenge that has yet to be resolved. Our previous works have progressively tackled this problem, from preliminary proposals exploring small language models tokenizers (DistillBERT) combined with logistic regression classifiers [37], to fine-tuned encoder-only models (BERT Multilingual) [38], and more recently, the exploration of decoder-only architectures adapted to perform multi-class classification [39] instead of the most complex multi-label classification. In this paper, we extend our previous works by exploring additional methods, including fine-tuning and comparing several encoder-only models, as well as testing in-context learning and fine-tuning of a decoder-only model. All methods are evaluated and compared over the same dataset.

Table 1
Multi-label text classification related works. The horizontal line distinguishes pre-LLM works from more recent LLM-based approaches.

Ref.	Year	Context	Labels	Samples	Methodology	Metrics
[12]	2022	Dialogue malevolence detection	18	8.4K	BERT-MCRF	F1-score: 0.49
[13]	2021	Cooperative patent categorization	600	70K	Transformed-based multi-task	F1-score: macro 0.40/micro 0.65
[14]	2019	EUROVOC labels	4.3K	57K	Fine-tuned BERT	F1-score: micro 0.73
[15]	2022	Scientific topics categorization	1.2K	186K	Hierarchical fine-tuned BERT	F1-score: macro 0.35/micro 0.53
[16]	2023	Clothing quality reviews	5	1.2K	Fined-tuned RoBERTa	F1-score: macro 0.87/micro 0.87
[17]	2023	Topics categorization from holidays rental platform reviews	239	120K	Fined-tuned BERT	mAP: macro 0.76/micro 0.93
[18]	2019	Hate speech detection	2, 5, 3	13K	Word n-gram and RFDT classifier	Average accuracy: 0.74
[19]	2022	Hate speech detection	7	46K	BERT trained from scratch	F1-score: 0.3 to 0.96
[24]	2021	Medical code categorization	18 to 923	40K to 200K	Fine-tuned BERT and RoBERTa	F1-score: macro 0.52 to 0.74/micro 0.68 to 0.81
[29]	2024	Aviation safety and autonomy	17	13.7	Fine-tuned GPT-3.5 and Mistral-7B	F1-score: macro 0.61/micro 0.74
[30]	2024	Disaster informatics	14, 2, 16	–	LORA-based fine-tuned Llama2	Training performance
[31]	2024	Patient comments classification	10	1089	Zero-shot and few-shot GPT-4 Turbo	F1-score: 0.76
[32]	2024	Unstructured electronic health records	83, 13, 36, 37	890K	Zero-shot and few-shot Llama-2 13B	F1-score: 0.75 to 0.86
[33]	2023	Multiple domains and datasets	6 to 90	Up to 58K	Label prompt learning	F1-score: 0.59 to 0.89
[35]	2024	Multiple mental health datasets	4 to 28	3.5K to 826K	Zero-shot LLMs	F1-score: 0.64
Ours	2025	Animated film production review notes	12	62K	Transfer learning, fine-tuned BERT family, QLoRa-based fine-tuned and one-shot Llama2-7B	F1-score: 0.89

3. Materials: problem formulation, data and processing

3.1. Problem statement

The formulation of the problem can be described as follows. The inputs consist of a collection of text data denoted as $X = \{x_1, x_2, \ldots, x_n\}$. Here, each x_i represents the ith text review note, which is typically a sequence of words of tokens, following pre-processing, from a total of n documents. The corresponding output is represented as a set of labels $Y = \{y_1, y_2, \ldots, y_n\}$, where each y_i is a binary vector of size m, with m being the total number of unique labels available in the dataset. In this context, m represents the total number of possible departments where the review note can be addressed (here, $m = 12$). The binary vector y_i signifies the presence (1) or absence (0) of each label for the corresponding text instance, allowing multiple '1's to reflect a multi-label situation. The primary objective is to learn a function $f : X \rightarrow \{0, 1\}^m$ such that, for any given input text x corresponding to a review note, the predicted output $\hat{y} = f(x)$ closely approximates the true label vector, effectively capturing the multi-dimensional label associations inherent in the text data.

3.2. Data description

Although it may seem that public access datasets related to film content are available, these only contain general comments made by critics and by the general public about already released films. These comments are not intended to improve particular aspects of the movie that has already been launched. This data is typically used for sentiment analysis [40], classifying comments as positive or negative.

However, the dataset used in this research contains much more detailed information on technical aspects of films during the production stage.

The data used in this work has been extracted from the film Luck, released worldwide in 2022. It is a production of Skydance Animation LLC, an American animation studio that is a division of Skydance Media. The studio is based in Los Angeles and Madrid; the Madrid branch was originally Ilion Animation Studios.

The data is generated during regular sessions with directors and supervisors organized by the Production Department. In these sessions, new and updated scenes and resources are reviewed and the Production Department transcribes the oral comments into a production tracking tool.

The different departments of the studio that attend are represented by a manager from that particular office. Each one must fill out a form, paying attention to the key aspects that refer to the specific work of their section.

Production Tracking is a method by which teams in the film industry can see how well the plan about the production of a movie has been executed. With the help of these tools, the Production Department

Table 2
Examples of review notes and the manually assigned labels.

Note text	Labels
Add more atmosphere to SL rock pillar so it is not as contrasty - at top of shot.	Lighting
We still need to work on the integration of the eyebrows 'whiskers'.	CFX & surfacing
Work in the change of attitude, we have to feel it.	Animation

checks how well the teams are doing their tasks and whether the jobs are up to the quality standards. In this case, Flow Production Tracking (formerly ShotGrid) [41] was used for tracking tasks and review notes.

In Table 2 it is possible to see some examples of real notes, as well as the department or departments it should be sent to. For the sake of brevity, only part of the note is shown, which is free text in English. Obviously, the form also collects information about: date, name of the film, production status, etc., although these fields are not used in this work.

This dataset is highly unstructured. It contains free-form text entered into the Production Tracking software by different people, and also in completely different styles and formats. Many of them do not have English as their mother tongue. This makes it very complex to apply traditional data mining techniques to extract useful information from these notes, so it is necessary to use advanced techniques based on natural language processing with language models, to focus on the content without depending too much on the structure.

Additionally, these notes contain confidential information. Character names and locations, especially in the case of unreleased films, are extremely sensitive. Neither artist names can be part of these datasets due to personal data protection laws.

The goal of the following processing phases of this information is creating an optimized dataset on which classification techniques can be successfully applied.

3.3. Data cleaning

The dataset was originally extracted from a movie Production Tracking software. There are several Production Tracking tools used in the film industry. To further extend this research as much as possible and be able apply these methods to information from all Production Tracking tools, a custom library was developed to serve as interface to extract the raw data from the original source. This way it can be adapted to the different Tracking software and be easily extended for future versions.

The dataset often contains unnecessary information, which can introduce noise and make classification difficult. To avoid this problem, a filtering process is carried out. This keeps only the information necessary to classify the notes [42].

To address potential issues with data confidentiality and proprietary rights, anonymization algorithms should be applied before using this production dataset outside of the study. There are libraries that provide various techniques such as k-anonymity, (α,k)-anonymity, l-diversity, entropy l-diversity, (c,l)-recursive diversity, basic β likeliness, enhancement *beta* likeliness, t-closeness, and disclosure privacy δ [43]. Additionally, all entity names in the dataset must be replaced using Named Entity Recognition (NER) algorithms [44], to respect intellectual property.

3.4. Label estimation

The original notes extracted from the Movie Production Tracking software contain many fields, but are not explicitly classified into the corresponding departments. To apply supervised machine learning techniques, it is necessary to have these notes labelled with the relevant departments. Given the large volume of notes and the time limitations associated with this task, manual labelling is infeasible. However, the information associated with the review notes contains additional fields, beyond the textual content, making it easier to infer the relevant departments. Specifically, each note includes a list of artists for email notification purposes, along with a catalogue of executed tasks. Leveraging this complementary data, labels can be estimated with sufficient reliability to allow the use of training algorithms. Artists usually belong to a department, and tasks are identified by types. Each task type is associated with the department that performs it. Using this relationship, the department label for each note can be inferred and added to the dataset.

Table 3
Distribution of review notes and target departments.

Department	Labelled notes
Animation	28.5%
CFX	28.1%
Lighting	19.5%
Final Layout (FL)	18.9%
Modelling	9.2%
Surfacing	7.4%
FX	3.7%
Layout	3.1%
Rigging	1.3%
Compo	1.2%
Crowds	1.2%
Matte	0.6%

During production, assignment of these notes has to happen right after the review meeting. At this point the note has not been notified to any artist and it does not contain any performed task to fix it. This extra information is usually added to the note as it is being addressed, usually several days after the review. This means this estimation method cannot be used. That is why a automated process to predict the labels identifying interested departments on a particular note has been proposed in this study.

The labelled dataset is then imported into data frames to speed up processing in later stages.

Each department operates with a different workflow dynamic. While certain departments, such as Animation, conduct internal reviews more frequently, others may carry out this work at a more advanced stage in fewer sessions, resulting in varying degrees of iteration. Consequently, the labelled dataset shows an imbalance, as certain departments, such as Animation or Lighting, have a higher number of labelled notes compared to others, such as Crowds or Rigging. Table 3 shows the distribution of the number of notes by department (label).

3.5. Tokenization

Transformer-based models cannot be trained with free-form textual data. Therefore, it is necessary to convert this textual information into a numerical representation suitable for feeding the model. This representation is typically in the form of tensor data and encapsulates the entire note text. This procedure is commonly known as tokenization or word embedding.

For this, different techniques can be used. Although the bag of words [45] technique is widely used, it has some inherent limitations as it fails to encapsulate the semantic nuances of words. Additionally, this coding method can be laborious due to the numerous variations of words in written text. In contrast, word2vec [46] addresses these shortcomings by grouping words with similar characteristics into shared embedding. Consequently, this approach makes it easier to understand and process text using machine learning algorithms, as sentences that convey similar meanings are represented in a similar way.

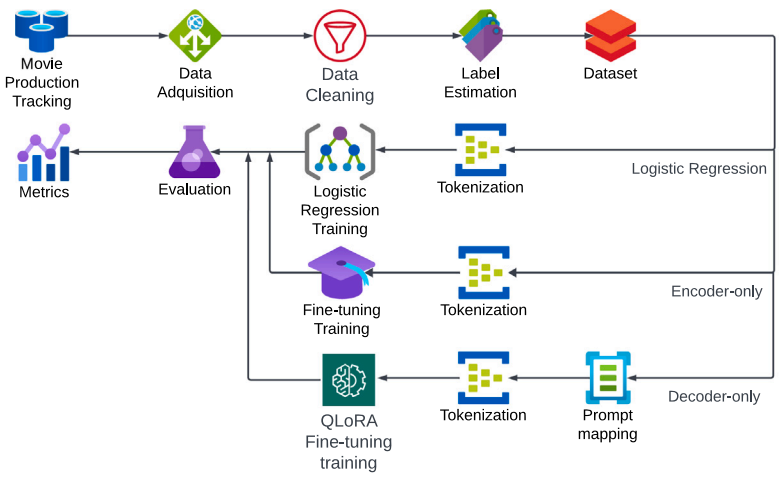

Fig. 1. Methodology for automatic distribution of animation review notes.

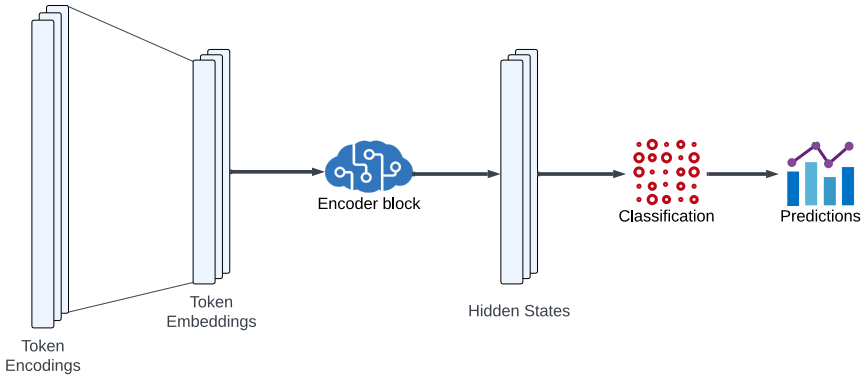

Fig. 2. Architecture of an encoder-only model for classification.

In the English language there are many words that reflect multiple meanings depending on their contextual use. Current tokenization algorithms differentiate between occurrences of the same word with different meanings, thus generating separate embeddings for each instance. Therefore this approach produces more accurate characteristic representations, improving the performance of classifiers that rely on these features as inputs.

Several tokenization strategies have been used in this research. The corresponding tokenizer is used when training a specific transformer model for classification. This way maximum compatibility is guaranteed.

These algorithms are part of the Subword' tokenization strategy. This approach is better than word and character-based tokenizers. Subword strategy does not divide words that are used frequently into smaller subwords. Instead, they only divide rare words, producing subwords that should be more meaningful.

Of those used in this work, BERT and DistilBERT tokenizers use WordPiece embeddings [47] with a 30K token vocabulary. RoBERTa and Llama 2 tokenizers use the Byte-pair encoding (BPE) algorithm [48, 49]. RoBERTa uses a variation called byte-level BPE [50] with a 50K subword units vocabulary while Llama 2 tokenizer uses the SentencePiece [51] implementation of BPE with a vocabulary size of 32K tokens.

4. Methodology

In this section the methods used, their description and the mathematical models and architectures that implement them are presented. In particular, two methods have been used: Encoder-only models and Decoder-only Large Language Models.

Fig. 1 represents an overview of the main processes followed to assign film review notes to the departments where animated films are developed. In the first line, the necessary data transformations are carried out (Section 3), to then apply each of the three proposed models (logistic regression, encoder-only models and decoder-only models) to those data.

The objective of the classification is to identify the departments interested in each particular review note. Using the textual content of the notes, the classifier is trained on the labelled data.

4.1. Encoder-only models

The classification architecture is built on an encoder-only transformer model. Encoder models use only the encoder of a Transformer model. At every stage, the attention layers have access to all the words in the original sentence. These models are commonly described as employing 'bidirectional' attention, as they account for the entire sentence, incorporating both preceding and subsequent words. These models are also called auto-encoding models. That is why Encoder-only models excel at tasks that require grasping the meaning of a sentence. The pre-training process for these models typically involves introducing some form of corruption to a given sentence, such as masking random words, and then training the model to reconstruct the original sentence. Examples of this family of models include BERT, DistilBERT, or RoBERTa.

An encoder-only model is used for tokenization and hidden states extraction, and serves as input to the classifier.

These hidden states are then classified in later stages to obtain the predictions of the department to which the note should be assigned.

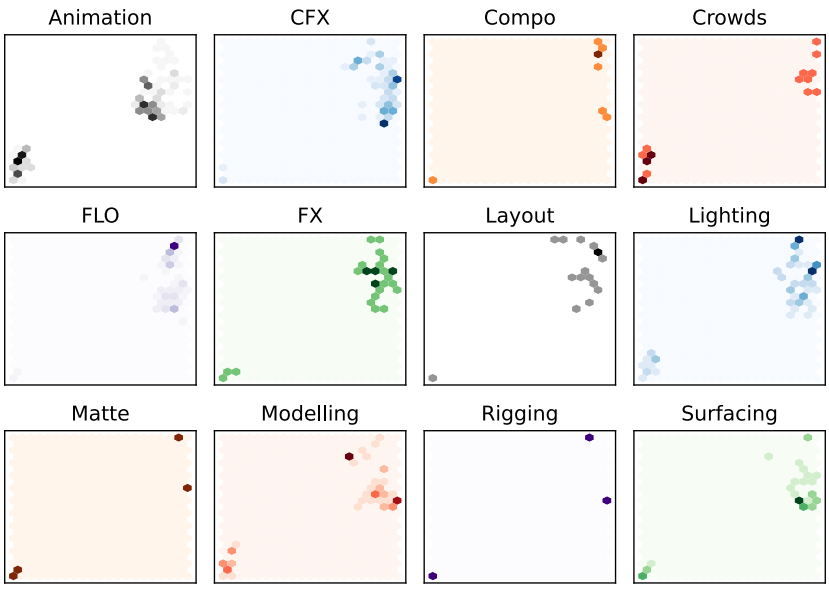

Fig. 3. Projection of hidden state features.

The architecture is shown in Fig. 2.

Each token serves as input to a hidden state which, in turn, functions as a feature to train the classifier, while maintaining the pre-trained state of the remaining components of the model. The model generates a 768-dimensional tensor as a hidden state for each input token. Typically only the hidden state corresponding to the beginning of the sequence token is used to obtain the final input for the classifier. That is, a unique 768-dimensional tensor is generated for each review note.

Since it is not possible to visualize the 768-dimensional input, Fig. 3 illustrates a 2D projection of this data using the Uniform Manifold Projection and Approximation technique [52]. The visualization shows different spatial partitions occupied by various departments, with departments sharing common tasks, such as Lighting and FX, exhibiting overlapping regions.

4.1.1. Transfer learning

As a first step, transfer learning using an encoder-only model has been used. The model used has been the DistilBERT variation [53].

A logistic regression model is added as classification head and trained [54] using the labelled inputs, which include both the hidden state and the previously calculated label. In this architecture, only the classification head is trained, corresponding to the logistic regression model, while the parameters of the layers of the encoder-only model, i.e., the left part shown in the architecture 2, remain fixed.

The limited memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) solver was used for the training. The model was trained using the Scikit-learn library with default parameters and run on a 3.00 GHz 18-core Intel i9 processor, equipped with 64 GB of RAM. The training duration on the CPU was 6.2 s. Given the multi-label problem of the dataset, one-vs-rest strategy was adopted. That is, a classifier is trained for each class, that in this particular case refers to each department of the animation studio. Since there are a total of 12 different labels in the dataset, the library creates 12 classifiers. Using the binary relevance method [55], it trains one binary classifier for each label, thus performing multi-label classification. The training was performed with a L2 penalty term for regularization and a tolerance for stopping criteria of 0.0001.

4.1.2. Fine-tuning encoder-only language models

The second method that has been proposed for the classification of movie text notes into the corresponding department is the use of fine-tuning to train encoder-only models for sequence classification.

In this case, the entire model shown in Fig. 2 is trained end-to-end. The starting point is a model pre-trained from a large general corpus. The training process then updates all the parameters of this pre-trained model using a smaller, domain-specific dataset. This dataset is the one that contains all movie review notes annotated with the automatic label estimation system described in Section 3.4. This process is called fine-tuning.

A general schema of the encoder elements can be seen in Fig. 4, showing the components of the model that will be trained with the fine-tuning process, without listing all the layers and functions inside each component. The BERT model contains a chain of 12 of those blocks connected. For full details of these types of model, the transformers original paper can be checked [56].

4.2. Decoder-only large language models

Decoder-only Large Language Models (LLM) main purpose is to generate new text taking into account some context provided. These models rely solely on the decoder component of a transformer architecture. At each step, the attention layers are restricted to accessing only the words that precede the current word in the sentence. Such models are commonly referred to as auto-regressive models. The pre-training process for decoder models typically focuses on predicting the next word in a sequence. These models are particularly well-suited for tasks that involve generating text. However, they can also be configured for other tasks like text classification. Examples of this family of models include GPT, Llama or Llama 2.

Decoder-only Large Language Models have been lately very successful and their use has recently increased in different domains. One of those models is Llama 2 [57], launched in 2023. This model has been released in several variations. The Llama 2-7B model is the one used in this study.

An overview of the Llama 2-7B decoder block components is shown in Fig. 5. Llama 2 uses the pre-normalization variation of the transformers model, using the Root Mean Square algorithm, that has shown good training stability and generalization. It uses a Grouped-query attention as the masked attention layer to speed up the inference process. Finally, the position-wise feed forward layer uses the SwiGLU activation function instead of ReLU or GeLU, since it has been found to perform better. Llama 2-7B is formed by 32 decoder blocks connected in sequence.

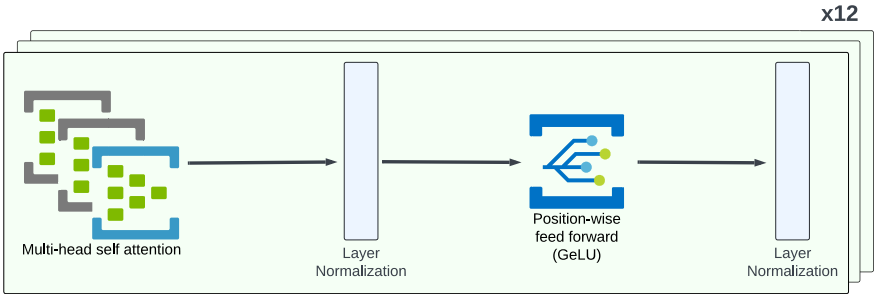

Fig. 4. BERT Encoder components.

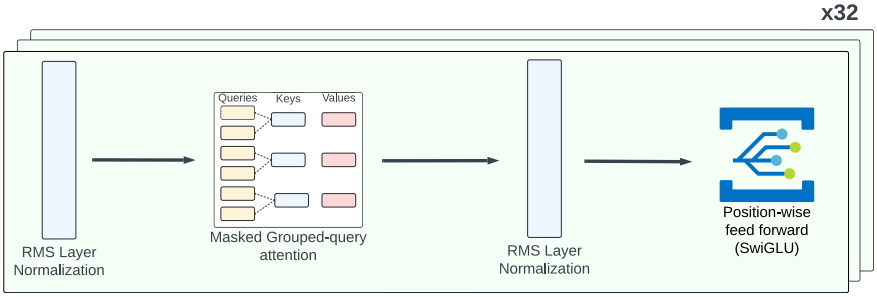

Fig. 5. Llama 2-7B decoder block components.

LLMs such as Llama 2 [57] are trained and evaluated through prompts that specify the task to be performed, with a structural approach. These training prompts follow the same format as those intended for implementation (prompt matching), ensuring optimal performance consistency. During this phase, the dataset is adapted to incorporate the note text within a prompt template, encompassing task descriptions and the list of departments for classification. With the Llama 2-7B model, a specific request format [57] is required.

Refining a language model through fine-tuning is relatively inexpensive in terms of computation compared to starting the training process all over again. However, for current large language models, such as Llama 2-7B, fine-tuning can require considerable computational costs that may become prohibitive except for large corporations.

To alleviate this computational burden, LoRA [58] was used, which allows the LLM to be trained locally on an RTX 3090 GPU. LoRA operates by approximating the fine-tuning weight update using a Rank Decomposition Matrix, which in turn decomposes into the product of two smaller matrices. This approach significantly reduces the number of trainable parameters, as LoRA exclusively optimizes the rank decomposition matrix instead of all LLM parameters. In this study, the Hugging Face Parameter Efficient Fine-Tuning (PEFT) library was used to train the model with LoRA, employing the post-weight update mechanism given by:

$$W_{ft} = W_{pt} + \Delta W = W_{pt} + \frac{\alpha}{r} AB \tag{1}$$

where W_{pt} are the weights of the pre-trained model, ΔW is the update of the model weights, AB is the Rank Decomposition Matrix, r is the dimension of the smaller matrices and α is a scaling factor used to balance the importance of the pre-trained weights and the updated ones.

To further improve the performance of this LLM model, QLoRA [59] was used in the training. QLoRA is a variant of LoRA in which the pre-trained model weights are loaded in a 4-bit quantized format, as opposed to the original 8-bit format. The QLoRA integration was achieved by using the bitsandbytes library along with PEFT.

5. Results and discussion

To evaluate and compare the three different proposed methodologies and ensure consistency of results, all models have been implemented and run on the same dataset. This dataset contains 33,785 text notes in total. The notes have been divided into 10% for tests (3379), and the rest for training and validation. 80% of those notes were used for training, a total of 24,324 labelled notes, and the rest (6082) for validation. The dataset has multiple labels, and each note can be allocated to multiple departments. The distribution of labels within the dataset can be seen in Table 3.

All experiments were run on the same hardware, consisting of a local computer equipped with an RTX 3090 GPU with 24 GB of VRAM with a i9 CPU at 3.00 GHz, using the dataset described in 3, with the same distribution of training, testing and validation subsets. Python 3.10.8 was used as the development language due to the availability of numerous open source libraries. Transformers [60] was used to interface with PyTorch with CUDA support and perform tokenization, inference and training tasks. Pandas [61] was also used for managing the dataset. Finally, Sklearn [62] was the library used for computing metrics from the results obtained from the different models and configurations. As a performance reference, fine-tuning the BERT model took 578 s for 5 epochs, for a resulting train loss of 0.13 and evaluation loss of 0.11.

As previously mentioned, given the multi-label nature of the classification, a One-vs-Rest approach was implemented for the transfer learning method. This methodology involves training a classifier for each class, where the classes in this dataset are the departments interested in the review text note.

Several metrics have been used to compare the different models and approaches. These metrics were evaluated through the SciKit-learn library. In the following equations, TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative.

Precision quantifies the proportion of True Positives relative to all detected Positives. This metric indicates the reliability of the system in assigning a specific department to a note.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

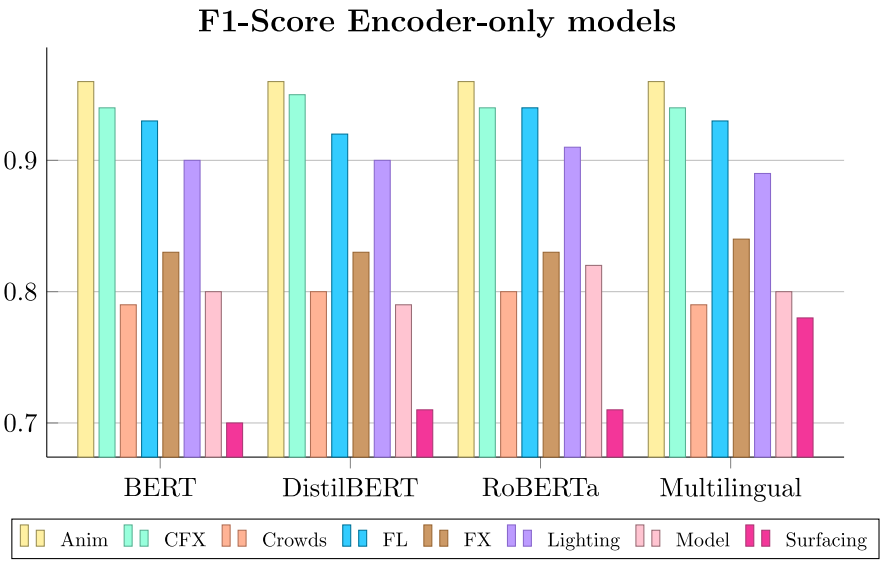

Fig. 6. F1-Score per label for the encoder-only language models.

Recall measures how often a machine learning model correctly identifies positive instances (true positives) from all the actual positive samples in the dataset. It is obtained by dividing the number of true positives by the number of positive instances of the dataset. Measures the effectiveness of the model.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

Accuracy shows the ratio of correct predictions over the total number of predictions made.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{4}$$

Finally, F1-score integrates precision and recall into a single metric to gain a better understanding of model performance.

$$\text{F1-Score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{5}$$

After testing several configurations, for the logistic regression classifier, DistilBERT was used to obtain the hidden states that serve as input to the classifier. The solver for this classifier was the Limited-Memory Broyden–Fletcher–Goldfarb–Shanno (lbfgs) from the SciKit-learn library.

To fine-tune the classifier with encoder-only models, several variations were tested. The BERT [63], DistilBERT [53], RoBERTa [64] and multilingual BERT [65] models have been chosen for testing. They were fine-tuned starting with 0.00002 as learning rate, using a weight decay of 0.01. The training was organized in batches of 8, obtaining a f1-score that was used to choose the best model (Fig. 6).

As it can be seen in this Fig. 6, all models perform similarly, although for some labels some models perform slightly better than others. Only a subset of the departments, the ones with more instances in the dataset, is represented in the figures for clarity. The worst result is obtained for the *Surfacing* department, which was entirely composed of Spanish workers. This may explain the better performance of the Multilingual model compared to the rest, since there could be Spanish words mixed with the text in English. Therefore, Multilingual has been selected in the final configuration of the encoder-only model for this particular dataset.

Regarding the configuration of the decoder-only LLM, both in-context learning and fine-tuning were evaluated. A prompt based on the structure recommended by Llama 2 was used. Explanatory text was introduced at the beginning of the prompt to define the task to be performed, detailing the list of departments into which to classify the text. Zero-shot inference showed a very low accuracy of 0.11, so one-shot inference was applied, adding an example of a note text for each of the departments. Few-shot inference was not used because, considering the large number of departments and the length of the notes (the examples provided in Table 2 are considerably shorter per space), the prompt would exceed the limits of the context length of the model, which is 4k for Llama 2.

The template used to implement one-shot learning is shown in Fig. 7, where {Example Note Text} refers to the text of a note from the training dataset, {departments} is replaced by a comma-separated list of the corresponding labels for that note, and {Input Note Text} is replaced by the note text from the test dataset that is being evaluated.

The fine-tuning started with an initial learning rate of 10^{-4}. The LoRA hyperparameters, r and α, were set to 16 and 64, respectively, as they commonly work very well with these datasets. The model covered a total of 3,540,389,888 parameters, of which only 39,976,960 were trainable. A paged optimizer was used throughout the training phase to avoid memory spikes and prevent out-of-memory errors. The prompt used for fine-tuning is the one used in one-shot learning, Fig. 7, with the examples removed.

The F1-score results obtained with both, the one-shot and fine-tuning methodologies applied to the Llama 2-7B model are shown in Fig. 8. Due to the confidentiality of the information, calculations must be performed locally, without using the cloud. This has limited the version of Llama2 used because of the limitations of the hardware available has prevented the use of larger versions of the software.

As it can be seen, in all cases fine-tuning obtains better results for this F1-score metric, i.e., higher values for all departments. Still, examination of the per-label metrics reveals that both perform better with departments with a broader spectrum acknowledged (e.g., Animation or Lighting) than with other more specialized ones (e.g. Final Layout). This is likely due to the model's prior exposure to information related to those fields within the pre-training datasets applied to the default base Llama2-7B model. Anyway, the fine-tuned model gives poor results for departments with limited number of sample, although still outperforming the one-shot approach.

5.1. Comparison of the three models

To compare the three methods of automatic classification of film review notes, the precision, recall, f1-score and accuracy metrics have

```
Categorize the movie review note text into one of the 8 categories:
CFX, Final Layout, Modelling, Layout, FX, Lighting, Compo, Animation

Examples:
Input:
{Example Note text}
### Response: {departments}
...
Input:
{Input Note text}
### Response:
```

Fig. 7. Template used to implement one-shot learning.

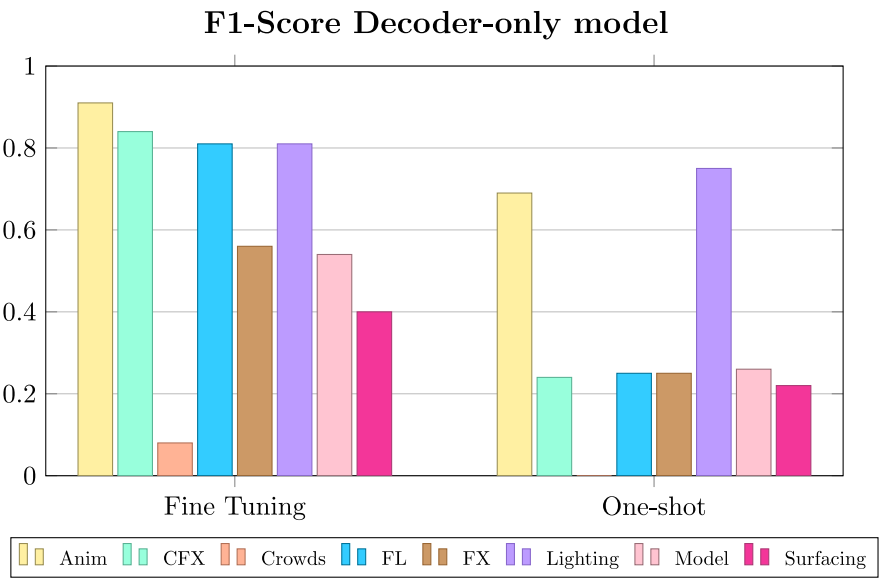

Fig. 8. F1-Score per label for decoder-only fine-tuned model.

Table 4
Weighted average metrics for classification methods.

Model	Precision	Recall	F1-score	Accuracy
Logistic regression	0.83	0.66	0.72	0.68
Encoder-only fine-tuning	0.90	0.88	0.89	0.83
Decoder-only	0.77	0.73	0.74	0.71

been evaluated.

Table 4 shows the numeric values of the weighted average metrics for all three classification methods, averaging the support-weight mean per label, giving more weight to classes with more samples.

From this data it can be seen that the best performance in classifying movie review notes is obtained with the use of fine-tuned encoder-only models. This confirms recent findings in comparative analysis like [66], where encoder-only models were found superior in performance over decoder-only models, that at the same time require less resources for analysis tasks like sentiment analysis. Decoder-only models typically require a larger number of parameters to obtain comparable results because they are designed to analyse text from left to right, which is more appropriate for generative tasks. However, encoder-only models take all surrounding context into account, making them excel at tasks like sentiment analysis or classification with fewer parameters than their decoder-only counterparts [67]. This could explain why encoder-only models show slightly better performance in this classification task, although decoder-only models could be improved by using a greater number of trainable parameters, but the hardware requirements may make them prohibitive on normal devices.

To illustrate some of these results, Fig. 9 shows the best F1-score value for each method.

From Fig. 9 it is possible to see that the fine-tuned encoder-only solution works very well for all departments, with high f1 even in departments with a low number of samples. On the other hand, the logistic regression and the decoder-only model worsen their performance especially in the departments with few notes, while they show competitive performance compared to the fine-tuned encoder-only model for the departments with many notes. Encoder-only models are very good at capturing the meaning of sentences and that is probably why, even with a few notes, they are able to extract representative features that allow the model to correctly predict the department in the dataset. In those departments with a large number of notes, Llama 2-7B shows very good results and since it is a prompt-based generative model, it could provide good options to extract information by querying the model, making it a very good solution if that flexibility is desired.

6. Conclusions and future research

In this article, several methods for classifying text notes, taken during animated film review meetings, have been proposed and compared. The final objective is to assign these notes to the departments in charge of addressing them. Real information provided by Skydance Animation Studios Madrid has been used. The main conclusion of this study is that it is possible to build an automated system with high levels of accuracy for this multi-label task. However, finding the right configuration of

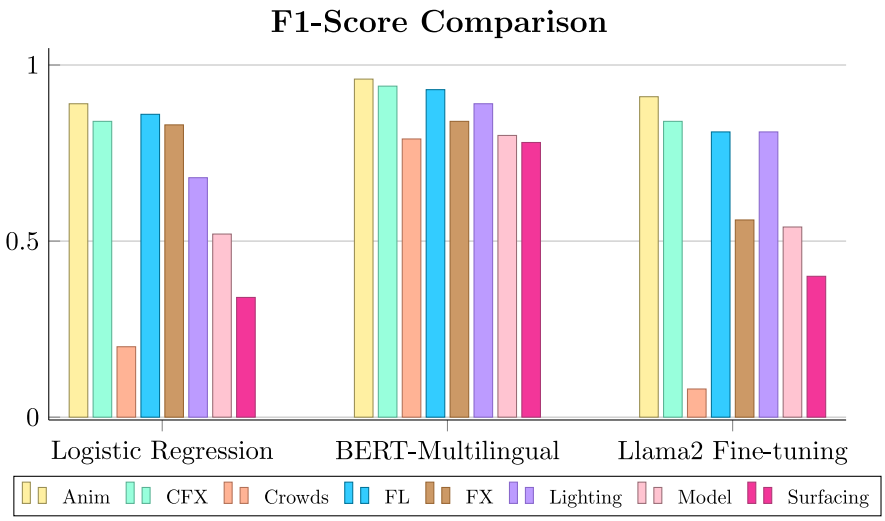

Fig. 9. F1-Score per label for all classification methods.

natural language processing models and tuning their parameters is not a simple task. It has been shown how different configuration options influence the final classification results.

On the other hand, the need to process real data to use it in training the models has also become clear. The three models compared were transfer learning with logistic regression, decoder-only LLM (specifically, Llama 2-7B) and fine-tuned encoder-only models. The latter have provided the best results in this specific application.

As future work, several lines are opened that would allow these models to be improved for this specific task. For example, LLMs like Llama 2 could be trained specifically for classification instead of using generative features. Other LLMs could also be explored such as Falcon, GPT-4, Mistral or the recently released model family Llama 3, trained with a corpus of 15T data token that has shown very good performance compared to other models with more parameters. The latter also increases its context length to 8K tokens, allowing more examples to be included. A future line of research involves contrasting the results obtained with datasets from other movies or even other studios to assess how generalizable these results are. Finally, the dataset can be improved by balancing it, since there are departments with fewer samples, which makes training more difficult and worsen the classification results.

CRediT authorship contribution statement

Diego Garcés: Writing – review & editing, Writing – original draft, Visualization, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Matilde Santos:** Writing – review & editing, Supervision. **David Fernández-Llorca:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

References

[1] A. Gasparetto, M. Marcuzzo, A. Zangari, A. Albarelli, A survey on text classification algorithms: From text to predictions, Information 13 (2) (2022) 83, http://dx.doi.org/10.3390/info13020083.

[2] A. Palanivinayagam, C.Z. El-Bayeh, R. Damaševičius, Twenty years of machine-learning-based text classification: A systematic review, Algorithms 16 (5) (2023) 236, http://dx.doi.org/10.3390/a16050236.

[3] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P.S. Yu, L. He, A survey on text classification: From traditional to deep learning, ACM Trans. Intell. Syst. Technol. 13 (2) (2022) 31:1–31:41, http://dx.doi.org/10.1145/3495162.

[4] M.J. Er, R. Venkatesan, N. Wang, An online universal classifier for binary, multi-class and multi-label classification, in: 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC, 2016, pp. 003701–003706, http://dx.doi.org/10.1109/SMC.2016.7844809.

[5] A. de Carvalho, A. Freitas, A tutorial on multi-label classification techniques, in: A. Abraham, A.E. Hassanien, V. Snášel (Eds.), Foundations of Computational Intelligence Volume 5, in: Studies in Computational Intelligence, vol. 205, Springer, Berlin, Heidelberg, 2009, pp. 177–195.

[6] A.N. Tarekegn, M. Ullah, F.A. Cheikh, Deep learning for multi-label learning: A comprehensive survey, 2024, arXiv:2401.16549.

[7] J. Liu, W.-C. Chang, Y. Wu, Y. Yang, Deep learning for extreme multi-label text classification, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 115–124, http://dx.doi.org/10.1145/3077136.3080834.

[8] R. Liu, W. Liang, W. Luo, Y. Song, H. Zhang, R. Xu, Y. Li, M. Liu, Recent advances in hierarchical multi-label text classification: A survey, 2023, arXiv:2307.16265.

[9] Y. Huang, B. Giledereli, A. Köksal, A. Özgür, E. Ozkirimli, Balancing methods for multi-label text classification with long-tailed class distribution, in: M.-F. Moens, X. Huang, L. Specia, S.W.-t. Yih (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 8153–8161, http://dx.doi.org/10.18653/v1/2021.emnlp-main.643.

[10] Y. Zhang, B. Jin, X. Chen, Y. Shen, Y. Zhang, Y. Meng, J. Han, Weakly supervised multi-label classification of full-text scientific papers, 2023, arXiv:2306.14003.

[11] X. Zhang, R. Abdelfattah, Y. Song, X. Wang, An effective approach for multi-label classification with missing labels, 2022, arXiv:2210.13651.

[12] Y. Zhang, P. Ren, W. Deng, Z. Chen, M. Rijke, Improving multi-label malevolence detection in dialogues through multi-faceted label correlation enhancement, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 2022, pp. 3543–3555, http://dx.doi.org/10.18653/v1/2022.acl-long.248.

[13] S.C. Pujari, A. Friedrich, J. Strötgen, A multi-task approach to neural multi-label hierarchical patent classification using transformers, in: Advances in Information Retrieval, Cham, 2021, pp. 513–528.

[14] I. Chalkidis, E. Fergadiotis, P. Malakasiotis, I. Androutsopoulos, Large-scale multi-label text classification on EU legislation, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 6314–6322, http://dx.doi.org/10.18653/v1/P19-1636.

[15] M. Sadat, C. Caragea, Hierarchical multi-label classification of scientific documents, in: Conference on Empirical Methods in Natural Language Processing, 2022, pp. 8923–8937.

[16] A. Alamsyah, N. Girawan, Improving clothing product quality and reducing waste based on consumer review using RoBERTa and BERTopic language model, Big Data Cogn. Comput. 7 (2023) 168, http://dx.doi.org/10.3390/bdcc7040168.

[17] F. Wang, M. Beladev, O. Kleinfeld, E. Frayerman, T. Shachar, E. Fainman, K.L. Assaraf, S. Mizrachi, B. Wang, Text2Topic: Multi-label text classification system for efficient topic detection in user generated content with zero-shot capabilities, 2023, arXiv:2310.14817.

[18] M.O. Ibrohim, I. Budi, Multi-label hate speech and abusive language detection in Indonesian Twitter, in: S.T. Roberts, J. Tetreault, V. Prabhakaran, Z. Waseem (Eds.), Proceedings of the Third Workshop on Abusive Language Online, Association for Computational Linguistics, Florence, Italy, 2019, pp. 46–57, http://dx.doi.org/10.18653/v1/W19-3506.

[19] B.D. Dirting, G.A. Chukwudebe, E.C. Nwokorie, I.I. Ayogu, Multi-label classification of hate speech severity on social media using BERT model, in: 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development, NIGERCON, 2022, pp. 1–5, http://dx.doi.org/10.1109/NIGERCON54645.2022.9803164.

[20] I. Mollas, Z. Chrysopoulou, S. Karlos, G. Tsoumakas, ETHOS: a multi-label hate speech detection dataset, Complex Intell. Syst. 8 (2022) 4663–4678, http://dx.doi.org/10.1007/s40747-021-00608-2.

[21] J. Lee, T. Lim, H. Lee, B. Jo, Y. Kim, H. Yoon, S.C. Han, K-MHaS: A multi-label hate speech detection dataset in Korean online news comment, 2022, arXiv:2208.10684.

[22] M.S. Hadj Ameur, H. Aliane, AraCOVID19-MFH: Arabic COVID-19 multi-label fake news & hate speech detection dataset, Procedia Comput. Sci. 189 (2021) 232–241, http://dx.doi.org/10.1016/j.procs.2021.05.086, AI in Computational Linguistics.

[23] A.E. Johnson, T.J. Pollard, L. Shen, L. wei H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L.A. Celi, R.G. Mark, MIMIC-III, a freely accessible critical care database, Sci. Data 3 (160035) (2016) http://dx.doi.org/10.1038/sdata.2016.35.

[24] V. Yogarajan, J. Montiel, T. Smith, B. Pfahringer, Transformers for multi-label classification of medical text: An empirical comparison, in: A. Tucker, P. Henriques Abreu, J. Cardoso, P. Pereira Rodrigues, D. Riaño (Eds.), Artificial Intelligence in Medicine, Springer International Publishing, Cham, 2021, pp. 114–123.

[25] S. Burkhardt, S. Kramer, A survey of multi-label topic models, SIGKDD Explor. Newsl. 21 (2) (2019) 61–79, http://dx.doi.org/10.1145/3373464.3373474.

[26] I. Chalkidis, M. Fergadiotis, S. Kotitsas, P. Malakasiotis, N. Aletras, I. Androutsopoulos, An empirical study on large-scale multi-label text classification including few and zero-shot labels, in: Conference on Empirical Methods in Natural Language Processing, 2020, pp. 7503–7515.

[27] X. Chen, J. Cheng, J. Liu, W. Xu, S. Hua, Z. Tang, V.S. Sheng, A survey of multi-label text classification based on deep learning, in: Artificial Intelligence and Security: 8th International Conference, ICAIS 2022, 2022, pp. 443–456.

[28] A. Edwards, J. Camacho-Collados, Language models for text classification: Is in-context learning enough? in: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC-COLING 2024, ELRA and ICCL, Torino, Italia, 2024, pp. 10058–10072.

[29] N.B. Niraula, S. Ayhan, B. Chidambaram, D. Whyatt, Multi-label classification with generative large language models, in: 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference, DASC, 2024, pp. 1–7.

[30] K. Yin, C. Liu, A. Mostafavi, X. Hu, CrisisSense-LLM: Instruction fine-tuned large language model for multi-label social media text classification in disaster informatics, 2024, arXiv:2406.15477.

[31] H. Sakai, S.S. Lam, M. Mikaeili, J. Bosire, F. Jovin, Large language models for patient comments multi-label classification, 2024, arXiv:2410.23528.

[32] D. Vithanage, C. Deng, L. Wang, M. Yin, M. Alkhalaf, Z. Zhang, Y. Zhu, A.C. Soewargo, P. Yu, Evaluating machine learning approaches for multi-label classification of unstructured electronic health records with a generative large language model, 2024, http://dx.doi.org/10.1101/2024.06.24.24309441, MedRxiv.

[33] R. Song, Z. Liu, X. Chen, H. An, Z. Zhang, X. Wang, H. Xu, Label prompt for multi-label text classification, Appl. Intell. 53 (8) (2023) 8761–8775, http://dx.doi.org/10.1007/s10489-022-03896-4.

[34] S.A. Tabatabaei, S. Fancher, M. Parsons, A. Askari, Can large language models serve as effective classifiers for hierarchical multi-label classification of scientific documents at industrial scale?, 2024, arXiv:2412.05137.

[35] A.A. Hassan, R.J. Hanafy, M.E. Fouda, Automated multi-label annotation for mental health illnesses using large language models, 2024, arXiv:2412.03796.

[36] M. Bucher, M. Martini, Fine-tuned 'small' LLMs (still) significantly outperform zero-shot generative AI models in text classification, 2024, http://dx.doi.org/10.48550/arXiv.2406.08660.

[37] D. Garcés, M. Santos, D. Fernández-Llorca, Text classification for automatic distribution of review notes in movie production, in: 18th Int. Conf. Soft Comp. Models Ind. Env. Apps., SOCO, 2023.

[38] D. Garcés, M. Santos, D. Fernández-Llorca, Language models for automatic distribution of review notes in movie production, in: Intelligent Data Engineering and Automated Learning, IDEAL, 2023.

[39] D. Garcés, M. Santos, D. Fernández-Llorca, Exploring large language models for automated review notes distribution in animation production, in: Intelligent Management of Data and Information in Decision Making: Proceedings of the 16th FLINS Conference on Computational Intelligence in Decision and Control & the 19th ISKE Conference on Intelligence Systems and Knowledge Engineering, FLINS-ISKE, 2024, pp. 153–160, http://dx.doi.org/10.1142/9789811294631_0020.

[40] L. Zhang, S. Wang, B. Liu, Deep learning for sentiment analysis : A survey, WIREs Data Min. Knowl. Discov. 8 (4) (2018) e1253.

[41] Inc.. Autodesk, Shine a light on your projects with flow production tracking (formerly ShotGrid), 2024, https://www.autodesk.com/campaigns/film-tv.

[42] V. Aubin, M. Mora, M. Santos, A new approach for writer verification based on segments of handwritten graphemes, Log. J. IGPL 30 (6) (2022) 965–978.

[43] J. Sáinz-Pardo Díaz, A. Lopez Garcia, A python library to check the level of anonymity of a dataset, Sci. Data 9 (2022).

[44] A. Roy, Recent trends in named entity recognition (NER), 2021, arXiv:arXiv:2101.11420.

[45] W. Qader, M. M. Ameen, B. Ahmed, An overview of bag of words; importance, implementation, applications, and challenges, in: Fifth International Engineering Conference on Developments in Civil & Computer Engineering Applications 2019, 2019, pp. 200–204.

[46] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, Vol. 26, 2013, pp. 3111–3119.

[47] Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, J. Dean, Google's neural machine translation system: Bridging the gap between human and machine translation, 2016, arXiv:1609.08144.

[48] P. Gage, A new algorithm for data compression, C Users J. 12 (2) (1994) 23–38.

[49] R. Sennrich, B. Haddow, A. Birch, Neural machine translation of rare words with subword units, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 1715–1725, http://dx.doi.org/10.18653/v1/P16-1162.

[50] C. Wang, K. Cho, J. Gu, Neural machine translation with byte-level subwords, 2019, arXiv:1909.03341.

[51] T. Kudo, J. Richardson, SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2018, pp. 66–71, http://dx.doi.org/10.18653/v1/D18-2012.

[52] L. McInnes, J. Healy, J. Melville, UMAP: Uniform manifold approximation and projection for dimension reduction, 2020, arXiv:arXiv:1802.03426v3.

[53] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter, 2019, arXiv:1910.01108.

[54] C. Prathibhamol, K. Jyothy, B. Noora, Multi label classification based on logistic regression (MLC-LR), 2016, pp. 2708–2712.

[55] M.-L. Zhang, Y.-K. Li, X.-Y. Liu, X. Geng, Binary relevance for multi-label learning: an overview, Front. Comput. Sci. 12 (2) (2018) 191–202, http://dx.doi.org/10.1007/s11704-017-7031-7.

[56] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38–45.

[57] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C.C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P.S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E.M. Smith, R. Subramanian, X.E. Tan, B. Tang, R. Taylor, A. Williams, J.X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom, Llama 2: Open foundation and fine-tuned chat models, 2023, arXiv:2307.09288.

[58] E.J. Hu, y. shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, in: International Conference on Learning Representations, 2022.

[59] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, QLoRA: Efficient finetuning of quantized LLMs, 2023, arXiv preprint arXiv:2305.14314.

[60] Inc.. Hugging Face, Hugging face transformers, 2024, https://github.com/huggingface/transformers.

[61] W. McKinney, Pandas: A Foundational Python Library for Data Analysis and Statistics, Vol. 14, 2011, pp. 1–9.

[62] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-Learn: Machine Learning in Python, Vol. 12, JMLR.org, 2011, pp. 2825–2830.
[63] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Pre-training of deep bidirectional transformers for language understanding, 2019, pp. 4171–4186, naacL-HLT.
[64] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, 2019, arXiv:1907.11692.
[65] T. Pires, E. Schlinger, D. Garrette, How multilingual is multilingual BERT? 2019, arXiv:1906.01502.
[66] A. Benayas, M. Sicilia, M. Mora-Cantallops, A comparative analysis of encoder only and decoder only models in intent classification and sentiment analysis: Navigating the trade-offs in model size and performance, 2024, http://dx.doi.org/10.21203/rs.3.rs-3865391/v1.
[67] S.H. Baskaran, A comparison of transformer and autoregressive LLM designs, Int. J. Res. Publ. Rev. 4 (11) (2023).

Diego Garces is currently a Staff Research Engineer at Skydance Animation. He is also Associate Professor at the University Complutense of Madrid (UCM), Spain and the University Center for Technology and Digital Art (U-Tad), Spain. He is a Ph.D. student at UCM in the field of Computer Science. He received his B.Sc. degree in Computer Science from the University of Zaragoza, Spain and the M.Sc. degree in Computer Engineering from the UCM, Spain.

He has worked in the entertainment industry for more than 20 years, working on blockbuster videogame titles and animated movies as part of technology leadership. His current research interests include the application of artificial intelligence, especially language models and generative AI to solve and improve the production process in the entertainment industry.

Matilde Santos is currently Full Professor for System Engineering and Automatic Control at the Computer Sciences Faculty, University Complutense of Madrid (UCM), Spain. She received her B.Sc. and M.Sc. degrees and her Ph.D. in Physics from the UCM, Spain. She is a member of the European Academy of Sciences and Arts. She belongs to the Council of the International Federation of Automatic Control (IFAC) and is the vice-president of the Spanish Committee of Automatic Control (CEA). She has published many papers in international scientific journals and several book chapters. She has supervised more than 15 PhDs. She has worked on several national, European and international research projects, leading some of them. She got several national and international awards. She serves as Associate Editor for different scientific journals. She is involved in different activities with social impact, being invited as a speaker at different event for science outreach and STEM initiatives.

Her current research interests include application of artificial intelligence techniques to different fields, mainly focus on intelligent control (fuzzy, neural networks, reinforcement learning), modelling and simulation, autonomous industrial vehicles, wind energy.

David Fernández Llorca is Scientific Officer at the European Commission - Joint Research Centre, and Full Professor at the University of Alcalá (Spain). He has co-authored over 180 publications, including journals, conferences, patents and science for policy reports. His research interests include trustworthy AI, AI evaluation, human-centred autonomous systems, predictive perception, and human–machine interaction.