From Bag-of-Words to Transformers: A Comparative Study for Text Classification in Healthcare Discussions in Social Media

Enrico De Santis, *Member, IEEE*, Alessio Martino, *Member, IEEE*, Francesca Ronci, and Antonello Rizzi, *Senior Member, IEEE*

***Abstract*—One notable paradigm shift in Natural Language Processing has been the introduction of Transformers, revolutionizing language modeling as Convolutional Neural Networks did for Computer Vision. The power of Transformers, along with many other innovative features, also lies in the integration of word embedding techniques, traditionally used to represent words in a text and to build classification systems directly. This study delves into the comparison of text representation techniques for classifying users who generate medical topic posts on Facebook discussion groups. Short and noisy social media texts in Italian pose challenges for user categorization. The study employs two datasets, one for word embedding model estimation and another comprising discussions from users. The main objective is to achieve optimal user categorization through different pre-processing and embedding techniques, aiming at high generalization performance despite class imbalance. The paper has a dual purpose, i.e., to build an effective classifier, ensuring accurate information dissemination in medical discussions and combating fake news, and to explore also the representational capabilities of various LLMs, especially concerning BERT, Mistral and GPT-4. The latter is investigated using the in-context learning approach. Finally, data visualization tools are used to evaluate the semantic embeddings with respect to the achieved performance. This investigation, focusing on classification performance, compares the classic BERT and several hybrid versions (i.e., employing different training strategies and approximate Support Vector Machines in the classification layer) against LLMs and several Bag-of-Words based embedding (notably, one of the earliest approaches in text classification). This research offers insights into the latest developments in language modeling, advancing in the field of text representation and its practical application for user classification within medical discussions.**

***Index Terms*—Embedding techniques, healthcare, natural language processing, social network analysis, text categorization, text mining, transformers, mistral, GPT-4, large language models.**

Manuscript received 16 September 2023; revised 26 February 2024; accepted 10 June 2024. *(Corresponding author: Enrico De Santis.)*

Enrico De Santis, Francesca Ronci, and Antonello Rizzi are with the Department of Information Engineering, Electronics and Telecommunications, University of Rome “La Sapienza”, 00184 Rome, Italy (e-mail: enrico.desantis@uniroma1.it; ronci.1404707@studenti.uniroma1.it; antonello.rizzi@uniroma1.it).

Alessio Martino is with the Department of Business and Management, LUISS University, 00197 Rome, Italy (e-mail: amartino@luiss.it).

Recommended for acceptance by M. Gong.

Digital Object Identifier 10.1109/TETCI.2024.3423444

I. Introduction

TECHNOLOGICAL evolution, particularly in the field of Artificial Intelligence, is not merely about individual advancements or methodologies. It encompasses a holistic integration of various technologies, facilitated by the “network effect” [1], which leverages ICT’s networking capabilities. This integration involves hardware acceleration through GPUs and TPUs, access to vast datasets and corpora for training procedures, and innovative algorithms for semantic representation, particularly in textual data. Natural Language Processing (NLP) has witnessed a paradigm shift similar to Computer Vision’s introduction of convolution in Convolutional Neural Networks. In NLP and text mining applications, this transformation emerged with the advent of Transformers and the attention mechanism, first proposed by Bengio et al. in 2014 for machine translation [2]. Vaswani et al.’s seminal work in 2017 [3] demonstrated how Transformers could overcome limitations in Recurrent Neural Networks, such as gradient problems, inefficiency in modeling long-range correlations, and lack of compatibility with parallel hardware. These Transformer architectures –within the so-called Large Language Model (LLM) family– exhibit the ability to handle the inherent complexity of natural language, which can be likened to a complex system [4], [5], [6]. But taking a step back in history, the Transformers were able to reach their maximum diffusion given the high performance also thanks to new neural methods of semantic-probabilistic embedding of the text and of the single words whose seed was thrown by Benjo himself in 2000 [7] and disclosed in his abilities by Mikolov et al. in 2013 [8]. In other words, modern Transformers, and all variants, rest on a solid semantic foundation provided by the techniques known as word embeddings, such as Word2Vec, which represent words as continuous vectors in a high-dimensional space. These embeddings capture semantic relationships between words, allowing for efficient semantic similarity computations. Unlike past traditional techniques that used sparse and discrete representations, word embeddings enable better generalization and analogical reasoning, as they encode semantic similarities and analogies through vector arithmetic. Such continuous representations marked a great step forward compared to traditional discrete techniques (e.g. Term Frequency-Inverse Term Frequency, TF-IDF) based on counting the frequency of appearance of words which took the first steps in the field of Information

Retrieval with the so-called Vector Space Model [9]. Word embedding techniques focus on the semantic representation of words in a local (semantic) perspective, while TF-IDF techniques work in a global perspective and produce sparse representations that perform poorly (also from a computational point of view) in representing very short texts such as posts on Social Networks (SNs). Similar considerations can be made on the Latent Semantic Analysis (LSA) technique [10] that, through a Singular Value Decomposition, represents words and documents in a lower-dimensional vector space, focusing on the most significant latent semantic dimensions. However, research works such as [11] showed that Skip-gram with Negative Sampling is very similar to a matrix decomposition involving the co-occurrence matrix (similar to Hyperspace Analogue to Language [12]) for word representations net to the intelligent sampling procedure. In this context, another approach known as GloVe (Global Vectors for Word Representation) was proposed in 2014 as an important alternative to other word embedding techniques like Word2Vec and FastText (2017) [13]. Unlike Word2Vec, which uses either Skip-gram or Continuous Bag-of-Words models based on local context, GloVe utilizes global word co-occurrence statistics from a large corpus. It constructs a word co-occurrence matrix, where each entry represents how often two words co-occur in a fixed context window across the entire corpus. The objective of GloVe is to learn word embeddings that capture the ratio of co-occurrence probabilities for different word pairs. So, GloVe can generate word representations that preserve both local context-based meaning and the overall distributional properties of words. It is important to note that the distributional hypothesis –similarity of meaning correlates with similarity of distribution– is at the basis of these embedding techniques, and it finds its principles in the Philosophical Investigations [14] of Ludwig Wittgenstein, in which he wrote: "*the meaning of a word is its use in the language*"; this approach was then introduced into linguistics (1950 s) by Z. S. Harris and J. R. Firth [5], [15]. Therefore, modern architectures for neural language processing through deep learning find their empirical foundations, first of all in the distributional hypothesis and compositionality of language and are then based on innovative methods of semantic representation of words through embedding and hierarchical processing. This is the case of the generative models (LLMs) such as the ChatGPT family [16] and deep architectures for textual classification such as the well-known Bidirectional Encoder Representations from Transformers (BERT) [17]. The latter –grounded on Transformer– is a powerful bidirectional language representation model pre-trained using a masked language model and next-sentence prediction objectives adopted mainly in text classification tasks. Instead, the former foresee a plethora of LLMs trained on gigantic corpora that can solve difficult tasks by showing "in-context" learning capabilities (e.g., GPT-3.5/4 or other open source models) [18], [19] or be adopted to generate powerful semantic vectors for text embedding. This is also the case, for example, of Mistral 7B LLM [20]. It is worth noting that in-context learning is a relatively new paradigm that allows LLM to learn tasks given only a few examples in the form of demonstration [21] using directly the prompt. This paradigm shift is considered a disruptive game changer within the NLP landscape [19].

After this brief report on the latest developments in the realm of language modeling, we can state that the main claim of the following study concerns the comparison of several text representation techniques in the context of classification of users who generate Facebook posts belonging to discussion groups (in Italian) on medical topics. Imagining –as just described– that the various text embedding and modeling techniques have settled over time and some techniques have become the basis for the generation of subsequent ones, e.g., word embedding as a basis for the construction of semantically relevant vectors for LLMs, this study intends to offer a comparison on a challenging problem since, as known, social network texts are short and often very noisy (and produced by non-specialist users). Furthermore, the Italian language makes the problem even more challenging given the scarcity of training material compared to the English language. Regarding the categorization of users based on text excerpts, this study utilizes two distinct datasets. The first dataset is employed to estimate word embedding models, while the second dataset comprises discussions from various users, grouped by topics (i.e., discussion groups). In this comparative analysis, the study evaluates different preprocessing and embedding techniques to achieve optimal user categorization, specifically aiming for the highest generalization performance by seeing the user categorization problem as a multi-class classification problem with imbalanced classes. Another important claim –within the comparative analysis– is to use data visualization tools to investigate the semantic embedding capabilities of users in relation to the performance of classification.

Specifically, this research delves into a comprehensive examination of the semantic representation capabilities of various word embedding techniques, along with certain pre-processing methods, which constitutes a significant focus of the study. So the nature of this paper is dual: from an application point of view, we want to build a performing classifier –within the Health Language Processing (HLP) field [22]– that can be part of a decision support system in the field of medical discussions, a challenging problem especially today when users usually get information on purely technical issues in medical settings on the Internet. This is important in monitoring discussions on medical topics to keep the danger of an infodemic [23], [24] or the spread of fake news and conspiracies [25]. From a scientific point of view, however, we are interested, as mentioned, in the representational capabilities of the new neural models, also i) in relation to the latest proposals regarding possible methods to enhance BERT, ii) adopting a powerful LLM such as Mistral 7B for the word embedding, iii) investigating in-context learning capabilities of GPT-4. Specifically, the following study extends our previous work [26], focusing on LLM, such as Mistral 7B, GPT-4 and BERT related to some of its variants, composing a single large comparison and targeting, with respect to the previous investigation, above all the classification performance. In this case, we will compare the classic versions of BERT with some hybrid solutions that present in the classification layer an approximate version –via Artificial Neural Networks– of Support Vector Machines (SVM) [27], [28].

This paper is organized as follows: in Section II the related works are revised while in Section III the adopted datasets are presented. Moreover, in Section IV the methodological framework is resumed. In Section V the results of the analysis are presented and discussed. Conclusions are drawn in Section VI.

II. Related Works

In recent years, the study and analysis of SNs have attracted considerable interest from the scientific community due to the overwhelming diffusion of digital communication through social platforms. Research reported in [29] highlights the potential of NLP in aiding *Computerized Clinical Decision Support*, benefiting both healthcare providers and the general public. This technology enables easy access to health-related information precisely when it is needed, facilitating decision-making processes. However, deep learning applications in biology and medicine, despite their intriguing possibilities, face certain challenges. For instance, the complexity of the data used in these applications is often not well understood, as mentioned in [30]. Additionally, deep learning models suffer from a lack of explainability, as reported in [31]. With the widespread usage of SNs, there is a valuable opportunity to examine how medical information is shared through these modern communication channels. In [32], the authors view SNs as a potentially valuable resource for detecting previously unknown drug side effects, as users often share valuable information about various aspects of their lives, including health-related matters. Another related study is presented in [33], where the authors analyze messages posted in an active Facebook diabetes group to identify key characteristics. Moreover, numerous investigations have focused on the use of word embedding in the field of biomedical natural language processing (also known as HLP). These studies explore large, unlabeled medical datasets like PubMed, clinical notes, Wikipedia, and news articles [34], [35]. Both qualitative assessments and quantitative measures reveal that word embedding models trained on medical corpora outperform pretrained ones such as GloVe [36] and Google News. Hence, studying how medical information is disseminated within these new communication frameworks is of utmost interest in technical literature. Furthermore, as already mentioned, the Transformers have proved to be a disruptive technology in text classification. By extension, this is true also in a medical context and healthcare environments. In [37], the authors aim to reduce the pressure of medical triage in the hospitals. Specifically, this paper proposes a medical triage system that could classify patients' questions or texts about their symptoms into several given categories. Authors, after building a real-world dataset including questions and answers with symptom tags, use BERT to give suggestions on which kind of consulting room patients could choose. BERT has also been used extensively during the COVID-19 pandemic to analyze posts on SNs. For example, in [38] the authors propose to adopt an unsupervised BERT model to classify sentiment categories (positive, neutral, and negative) and a TF-IDF model to summarize the topics of posts pertaining to Sina Weibo, a popular Chinese social media. Authors claim that fine-tuned BERT conducts sentiment classification with considerable accuracy. The investigation proposed in [39], instead, claims that not much effort is done to use fine-tuned BERT in Covid-19 datasets containing news blogs, posts, and tweets, also related to hate speech and fake news. The study in [40] proposes a technique to automatically distinguish posts that self-report the user's exact age from those that do not use, both for Twitter and Reddit posts data and comparing both BERT and RoBERTa [41]. In [42] RoBERTa to classify five prominent kinds of mental illnesses (depression, anxiety, bipolar disorder, ADHD and PTSD) by analyzing unstructured user data on Reddit, proposing also a high-quality dataset to drive research on this topic. On the other hand, authors in [43] face the problem of early depression detection in SNs in the field of psychology. Specifically, the study proposes a depression analysis and suicidal ideation detection system, for predicting the suicidal acts, using BERT and multiple instance learning approaches. Transformer-based architectures are relatively new and high-performing. Nevertheless, many researchers are focused on enhancing these architectures or even integrating them with more "mature" and well-established methodologies. For instance, the authors in [44] propose to evaluate humor in edited news headlines by using two hybrid systems, "BERT+EDA" – a fine-tuned BERT model with data augmentation, and "BERT+NB-SVM" – a hybrid model combining BERT and Naive Bayes with SVM. BERT+NB-SVM outperforms BERT+EDA in both subtasks. These hybrid systems have intrigued many researchers, so much so that authors in [45] and [46] have found very interesting strategies to represent a SVM using neural networks. Under certain specific conditions, they achieve a true conceptual similarity between the two methodologies, and by leveraging the strengths of neural networks, they still achieve good performance during testing with a significant reduction in evaluation times. Finally, as the family of LLMs are concerned, in [47] Mistral 7B is compared with other LLMs in several tasks involving text embedding, while [48] explores the use of GPT-4 for summarization tasks in the medical field, specifically focusing on medical dialogue summarization.

III. Datasets

In this study, two datasets were utilized. The first dataset comprises leaflets of medical products and was employed to train language models using neural word embedding techniques. These leaflets were obtained from a public Italian website called *MyPersonalTrainer*,[1] which contains various medical information, including a collection of medicine leaflets distributed in Italy via the *Agenzia Italiana del Farmaco*[2] website.

The second dataset consists of public posts extracted from three Facebook discussion groups [26]:

- *Dieta e diabete di tipo2 DMT2;*[3]

[1] [Online]. Available: https://www.my-personaltrainer.it/Foglietti-illustrativi/ [in Italian]

[2] Transl: *Italian Medicines Agency*

[3] Transl: *Diet and type 2 diabetes DMT2*

TABLE I
EXAMPLE OF FACEBOOK POSTS ON MEDICAL TOPICS

Groups	Posts
Group 1 : Dieta e diabete di tipo 2	Ho bisogno di avere dei compagni di viaggio sopratuttoDi poter consigliarmi con voi quando sono giù per i valori alti!!! Grazie e siete grandi con il supporto che posso avere !!!@Grande il Dottor ****!!!!
	salve a tutti. una domanda per chi è del gruppo 0. l'echinacea fa male, con cosa la sostituite per aumentare le difese immunitarie in inverno? grazie mille e buona serata :)
	Scusa Enrica lo so che non centra niente con il gruppo, sai dirmi se c'è un video del dott. **** sulla dieta x diverticoli?Grazie mille!
Group 2 : Tumore al colon-retto, restiamo vicini	Buonasera a tutti! Noi, dopo l'ultima operazione a dicembre, siamo stati circa dieci giorni fa dall'oncologo. Scrivo noi, ma è mia mamma di 52 anni che sta combattendo. ****, con cui mia mamma aveva già fatto radio e chemio dopo un primo intervento nel 2017, ha detto che non si possono fare altre terapie (in qualche modo eravamo "preparati" perché avevo chiesto un consulto anche al Prof. ****). Ha finito la chemio verso la metà di ottobre, l'11 novembre stava già male.
	Qualcuno con la stadiazione pT3N1 G2?
	TAC con liquido di contrasto per verifica progressi fatti, oggi :Tutta la massa e le lesioni epatiche ridotte del 40% con appena 4 cicli di chemioterapia. Analisi perfette, marks tumorali notevolmente ridotti... Grandissimo risultato 😜 😜
Group 3 : Italia-glioblastoma multiforme-cancro al cervello	A rieccomi!!! 🙂 Hanno riscontrato a mio marito una mucosite di III grado che gli ha intaccato sia la cavità orale che quella dell'ano.Sta malissimo e nonostante gli sciacqui e le varie creme la situazione peggiora e basta.Domani ha l'8 chemio e nn oso immaginare come starà nei prox giorni 😟 Passeremo in reparto e chiederemo aiuto agli oncologi.. voi nel frattempo cosa consigliate?Cosa posso dargli da mangiare? E per dargli un po' di sollievo??
	gliosarcomaBuongiorno a tutti, ho trovato un sacco di materiale che riguarda il GBM ma poco o niente su gliosarcomi..mi potete indirizzare ? Mia suocera è stata operata ad agosto e ora sta facendo radio e chemio.. grazie
	Protocollo ****Inutile stare qui a spiegare la situazione clinica di mio papà, dico solamente che è stato operato circa 20 gg fà e non è stato asportato totalmente. Adesso dovrebbe iniziare radio e chemio ma alcuni amici fidatissimi mi consigliano di seguire "il protocollo Puccio" un chimico ricercatore palermitano. Essendo un pò scettico chiedo a voi allegando il sito cosa ne pensate, sono un neofita di tutto ciò perciò mi affido a voi per un parere.[link]

Asterisks in the text are not part of the original posts but have been placed for privacy reasons.

TABLE II
DATASET FOR CLASSIFICATION TASK [26]

Group #	Group name	Authors	Posts
1	Dieta e diabete di tipo 2	478	957
2	Tumore al colon-retto, restiamo vicini	272	1262
3	Italia-glioblastoma multiforme-cancro al cervello	62	305
	Total	812	2524

- *Tumore al colon-retto, restiamo vicini;*[4]
- *Italia - Glioblastoma Multiforme - cancro al cervello.*[5]

These three groups were chosen based on the following rationale: the last two groups collected experiences of patients with different types of cancer, which provided rather similar contexts for analysis. On the other hand, the first group focused on a specific diet proposed for diabetic patients, offering a completely different context compared to the other two groups. It is important to note that no assessment was made regarding the medical quality of the information shared within the discussions.

Originally, the dataset contained 5855 posts authored by 1045 distinct individuals, resulting in an average of 5.6 posts per author. To ensure data privacy, the dataset was anonymized before analysis, and a preliminary data-cleaning process was performed. Posts containing only emojis or a small number of words, such as greetings or brief messages, were discarded. Additionally, posts with fewer than 14 tokens were removed to avoid empty or uninformative messages after subsequent pre-processing steps – please refer to [26] for more details. After this data cleaning phase, the final dataset consisted of 2524 posts written by 812 distinct authors, with an average of 3.1 posts per author. A set of example of posts for each groups is reported in Table I.

Table II provides an overview of the distribution of authors and posts in each group, highlighting the imbalance in the dataset:

IV. METHODOLOGY

This study addresses the problem of classifying users who write posts on a medical topic discussion group in a SN. In the following, we formally introduce the problem (cf. [26]). Let $U = \{u_1, u_2, \ldots, u_M\}$ be the set of users, each of which is associated to a set of posts (documents) $P_j = \{p_{j1}, p_{j2}, \ldots, p_{jK(u_j)}\}$, where $j = 1, 2, \ldots, M$, and $K(u_j)$ represents the number of posts written by the j-th user. Each post $p_{jK(u_j)}$ consists of a set of W words $S = \{w_1, w_2, \ldots, w_W\}$. The main objective is to develop a model $\mathcal{M}$ using a classifier as a predictive model. The free parameters of the model are learned by providing a set of $\langle u, c\rangle$ pairs, where $c \subseteq \mathcal{C} = \{c_1, c_2, \ldots, c_l\}$, and the latter being the set of class labels, to a training algorithm. In other words, the training process enables learning a decision function f that takes an input u and returns a predicted class label $\hat{c}$, denoted as $\hat{c} = f(u, \theta)$, where θ represents the set of free parameters of the model $\mathcal{M}$. The dataset of users $\langle u, c\rangle$ is then divided into two disjoint sets: the training set $\mathcal{S}_{\text{tr}}$ and the test set $\mathcal{S}_{\text{ts}}$, respectively employed to train the model and to test its generalization capabilities with 80% and 20% of the available patterns. To optimize the underlying hyperparameters of the classification algorithm, a validation set $\mathcal{S}_{\text{vs}} \subset \mathcal{S}_{\text{tr}}$ is used.

To employ a machine learning algorithm, it is necessary to represent the text of posts and users adequately. This is achieved by mapping documents (and users) to real-valued vectors using an appropriate embedding function $\Gamma : \mathcal{U} \rightarrow \mathbb{R}^n$, where $\mathcal{U}$ represents the space of users, and n is the dimension of the embedding vector. To accomplish this, an embedding function $\Lambda : \mathcal{W} \rightarrow \mathbb{R}^n$ is defined for words w in each post, where $\mathcal{W}$ is the space of all unique words (i.e., the vocabulary) in the dataset of posts.

Since the focus of this investigation is on classifying users rather than individual posts, the user embedding Γ (user-vector v_u) is constructed as follows:

$$v_u = \frac{1}{k(u)} \sum_{j=1}^{k(u)} v_j, \tag{1}$$

where v_j is the embedding vector for the post p_j, evaluated, in turn, as:

$$v_j = \frac{1}{W} \sum_{w=1}^{W} v_w, \tag{2}$$

where v_w is the embedding vector for the word w obtained trough the representation function Λ.

[4] Transl: *Colorectal cancer, we stay close*

[5] Transl: *Italy - Glioblastoma Multiforme - brain cancer*

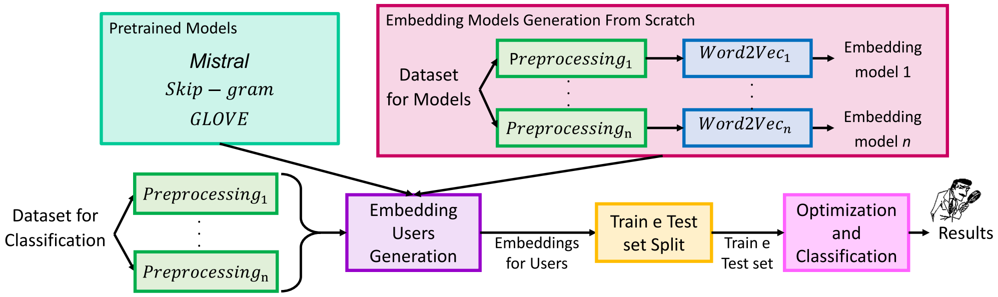


Fig. 1. High-level block diagram sketching the user classification task based on neural word embedding [26].

Therefore, it is crucial to have an effective embedding function Λ that preserves semantic relationships between word-vectors. This function can be obtained using a neural word embedding algorithm, which will be discussed in Section IV-B.

Fig. 1 provides an overview of the general approach, outlining the user classification task based on neural word embedding. Prior to the user classification module, there is a module responsible for generating the user's embedding vector using an appropriate procedure, which, in this case, utilizes a neural word embedding algorithm. This module can be fed with embedding vectors obtained from either a pre-trained model or a model trained "from scratch" using the leaflets corpus, as explained in Section IV-B. However, for both datasets (used for training user classification and training word embedding "from scratch"), a suitable pre-processing procedure is necessary. Different pre-processing and data transformation strategies are explored for the word embedding trained from scratch, leading to different sets of word-vectors, which will be the primary basis for comparison.

A. Pre-Processing

The dataset undergoes several pre-processing steps, detailed in [26]:

- Customized word-level Tokenization;
- Simple Tokenization;
- N-grams Generation with $N = \{1, 2, 3\}$;
- Data Cleaning and Emoji Removal.

Depending on whether a given step is employed or not in the pre-processing stage, it is possible to draw different combinations, yielding in turn different Pre-Processed Dataset (PPD):

PPD1: Customized Tokenization and Emoji Removal are applied;
PPD2: Customized Tokenization, N-grams Generation and Emoji Removal are applied;
PPD3: Customized Tokenization, Data Cleaning and Emoji Removal are applied;
PPD4: Customized Tokenization, N-grams Generation, Data Cleaning and Emoji Removal are applied;
PPD5: Simple Tokenization, N-grams Generation and Emoji Removal are applied;
PPD6: Simple Tokenization, N-grams Generation, Data Cleaning and Emoji Removal are applied;
PPD9: only Emoji Removal is applied (it applies for BERT, Mistral and GPT-4).

Additionally, two more datasets are created as follows:

PPD7: as a merging of PPD1 and PPD2;
PPD8: as a merging of PPD3 and PPD4.

yielding a total of 9 different types of pre-processing stages (hence, resulting datasets) to be fed to the different embedding strategies.

It is important to highlight that the pre-processing technique used for $\mathcal{S}_{\text{tr}}$ is also applied to $\mathcal{S}_{\text{ts}}$ to ensure uniformity and fairness in the comparison between the training and testing phases. The aim is to maintain consistency in the data preparation process for both sets. However, there is one exception to this rule. For pretrained models, a training phase is not required. Therefore, the PPD1 pre-processing technique will be applied to the test dataset in the case of Word2Vec pretrained models, while PPD9 pre-processing will be applied for BERT, Mistral and GPT-4 models. Thanks to their peculiar architectures, BERT and Mistral are able to generate contextual embeddings by taking an entire sequence of words as input, including punctuation. This approach allows us to utilize the pretrained model's knowledge while still preparing the test dataset appropriately to evaluate its performance effectively.

B. Word Embedding Models

Word embedding models are used to generate a hopefully good representation function Λ of the tokens (words) that can explicit a possibly valid semantics to solve the classification problem. As already noted, two strategies for neural word embedding are experimented within this general scheme –see Fig. 1. The first method uses pre-trained language models, while the second one consists of a training stage performed "from scratch" using the leaflets dataset –see Section III. The pre-trained word embedding models used in this work are Skip-gram and GloVe, whereas the word embedding models from scratch were created through the Word2Vec algorithm [8] –see [26] for more details.

Given the 9 PPDs described in Section IV-A and the 3 candidate neural word embedding models, it is possible to train the following models, already introduced in [26]:

Mdl1: Word2Vec trained with PPD1;
Mdl2: Word2Vec trained with PPD2;

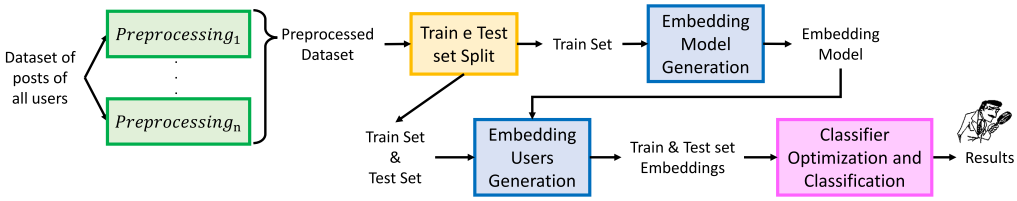

Dataset of posts of all users
$Preprocessing_1$
$Preprocessing_n$
Preprocessed Dataset
Train e Test set Split
Train Set
Embedding Model Generation
Embedding Model
Train Set & Test Set
Embedding Users Generation
Train & Test set Embeddings
Classifier Optimization and Classification
Results

Fig. 2. High-level block diagram sketching the user classification task based on traditional BoW-based embedding [26].

Mdl3: Word2Vec trained with PPD7;
Mdl4: Word2Vec trained with PPD5;
Mdl5: pre-trained vanilla Skip-gram model;
Mdl6: pre-trained vanilla GloVe model;
Mdl7: Word2Vec trained with PPD3;
Mdl8: Word2Vec trained with PPD4;
Mdl9: Word2Vec trained with PPD8;
Mdl10: Word2Vec trained with PPD6.

By training and running these models, the vector representation of the words of the posts is obtained, from which the embedding vectors for Facebook users are derived, as formally described in Section IV. Recall Fig. 1 for an overview.

Since the embedding vector is a particular type of representation (mainly driven by the embedding model under analysis), a new representation space can be obtained by concatenating word embedding vectors computed with different models. The newly merged word embedding vectors for the users are [26]:

Mdl11: by juxtaposition of the user's embedding for Mdl3 and Mdl5;
Mdl12: by juxtaposition of the user's embedding for Mdl3 and Mdl6;
Mdl13: by juxtaposition of the user's embedding for Mdl4 and Mdl5;
Mdl14: by juxtaposition of the user's embedding for Mdl4 and Mdl6.

The rationale behind this merging between a pretrained model (Mdl5 or Mdl6) and an ad-hoc model trained on the leaflet corpus (Mdl4 or Mdl3) is to include in our investigation whether the mixture of both worlds can be fruitful in solving the classification problem for the two families of neural models, if considered alone.

C. Traditional Approaches

The process underlying the traditional approach is outlined in Fig. 2. It is based on the Bag-of-Words (BoW) matrix and its LSA representation [10], [49]. Unlike neural word embedding methods, which usually start with the words dataset, the starting point is the dataset of users' posts, which is immediately split into a training set and a test set. Both datasets undergo separate pre-processing, resulting in post's PPD (see Section IV-A), which then undergoes a sequence of transformations, including BoW, TF-IDF, and LSA –see [26] for more details. In order to generate different BoW models, four types of pre-processing strategies are applied:

Mdl15: BoW–TF-IDF–LSA starting from PPD1;
Mdl16: BoW–TF-IDF–LSA starting from PPD2;
Mdl17: BoW–TF-IDF–LSA starting from PPD3;
Mdl18: BoW–TF-IDF–LSA starting from PPD4.

D. Support Vectors Machine for Text Classification

The SVM [50] is a supervised learning technique that focuses on classifying data into two or more categories. The effectiveness of SVM in handling a wide range of complex problems, its ability to generalize, and its high flexibility make it one of the preferred choices for classification in various fields, such as image recognition, text analysis, bioinformatics, and more. In its traditional form, SVM is a non-neural algorithm but under certain conditions and with some approximation, it can be represented in a neural form as explained in [45], [46]. One common approach is to reformulate the SVM objective function as a differentiable loss function that can be optimized using gradient-based methods, such as stochastic gradient descent. By doing so, SVM-like behavior can be achieved within a neural network architecture. These neural network-based approximations of SVM are often referred to as "Quasi-SVM" or "Neural SVM". This family of techniques potentially benefits from the scalability and flexibility of neural networks while retaining some properties of SVM. It is important to note that while neural SVM approximations can be useful in certain scenarios, they might not fully replicate the exact behavior of traditional SVM in all cases.

In this work, we propose the study and evaluation of two neural SVMs:

- *Quasi-SVM* block: proposed by the Keras developers[6]
- *Kernel regularizer* block: formulated by us.

The *Quasi-SVM* block performs a transformation on its inputs, projecting them into a feature space with a specific number of dimensions. The purpose of this transformation is to approximate shift-invariant kernels, which are kernel functions that exhibit the property of invariance to shifts in the input space. In other words, the kernel function's value between two points x and y only depends on their distance, represented by the function $k(x-y)$. To achieve this, the layer applies a mapping that allows it to emulate popular Radial Basis Functions (RBF) as Gaussian kernel, which is a shift-invariant kernel [51]. Finally, this layer is utilized to *kernelize* linear models, wherein it applies a non-linear transformation (the layer itself) to the input features. The transformed features are then used to train a linear model.

[6] [Online]. Available: https://keras.io/examples/keras_recipes/quasi_svm/

The outcome of this combination depends on the loss function of the model.

The *Kernel regularizer* block is a neural network constructed "manually" to mimic the behavior of a SVM. Some researchers have shown that it is achievable by overlaying one or more hidden layers with suitable activation functions to introduce non-linearity and regularization factors As a result, they construct a classifier block that resembles a SVM by stacking multiple dense layers and including ℓ_2 regularization and a linear activation function in the final layer [45], [46]. In essence, these layers facilitate the creation of models that mimic the behavior of kernel-based methods in combination with linear models.

Since we are focused on SVM, the hinge loss function is used for both Quasi-SVM and Kernel regularizer, so the resulting model is equivalent (with some degree of approximation) to kernel SVMs.

E. LLMs for Text Classification: BERT, Mistral and GPT-4

The evolution of NLP has been significantly shaped by the development of LLMs, grounded on Transformer architecture, such as BERT, Mistral, and GPT-4. BERT marks a departure from traditional unidirectional language models by employing a bidirectional training method to better understand word context. Following BERT, Mistral extends these capabilities with high-quality multilingual support. GPT-4 Turbo, also capable of working in a multilingual setting, further innovates with its exceptional in-context learning abilities, demonstrating adaptability across Zero-shot and Few-shot scenarios. Each model leverages transfer learning to fine-tune pre-trained weights for diverse NLP tasks, showcasing their unique strengths in text classification and language analysis. The original Transformer architecture [3], which is the foundation of BERT and other LLMs, consists of an Encoder-Decoder structure. However, in BERT, only the Encoder part is used, as it is designed for tasks that involve understanding the meaning of sentences, rather than generating new text. The Encoder is composed of multiple layers, each containing Multi-Head Self-Attention mechanisms and Feed-Forward neural networks [17]. The Self-Attention mechanism allows BERT (and other LLMs) to weigh the importance of different words in a sentence when encoding each word's representation. It computes attention scores between each word and all other words in the sentence, considering both previous and subsequent words. This enables the model to have a contextual understanding of the entire sentence while encoding a particular word.

Formally, given a sequence of input tokens, denoted as $X = \{x_1, x_2, \ldots, x_n\}$, where x_i represents the i-th token, LLMs computes the hidden representation $H = \{h_1, h_2, \ldots, h_n\}$, where h_i is the representation of x_i learned by the models.

The Self-Attention mechanism in LLMs is defined as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{3}$$

where Q, K, and V are the Query, Key, and Value matrices, respectively. These matrices are obtained by linearly projecting the input embeddings, and d_k is the dimension of the Key vectors. Moreover, the Multi-Head Attention mechanism combines multiple Self-Attention *Heads* to capture different aspects of contextual information. The outputs of the Attention Heads are concatenated and linearly transformed to produce the final Attention output for each word. BERT further incorporates a Masked Language Model (MLM) and Next Sentence Predictions (NSP) objectives during pre-training. The MLM involves randomly masking some tokens in the input during training and tasking the model with predicting the masked tokens. This encourages BERT to learn bidirectional representations and enhances its ability to understand the context even in the absence of certain words. On the other hand, NSP is used to learn the ability to recognize if two input sentences are contextually and conceptually consecutive or not, within a corpus.

In the end, BERT is a powerful Transformer-based model that learns bidirectional representations by leveraging Self-Attention mechanisms and MLM and NSP objectives. Its pre-trained weights can be fine-tuned on various downstream NLP tasks, making it a versatile tool for natural language understanding, classification problems, question answering and named-entity recognition tasks. In each case, a Classification layer is added to the Encoder block's output. In summary, fine-tuning enables BERT to serve as a starting point for various end-task models by adapting its parameters and weights for specific applications. This technique is the aforementioned "Transfer Learning" approach, which uses acquired knowledge to solve new problems.

Mistral is a highly performing multilingual decoder-only LLM, which also supports the Italian language. One of its distinctive features is its context window, which spans up to 8 K tokens. This signifies its ability to effectively analyze documents containing a maximum of 8 K tokens, thereby capturing a broad contextual understanding crucial for tasks such as document classification. Mistral, particularly in its *Mistral-embed* configuration, proves to be a potent tool for generating high-dimensional embedding vectors conducive to document classification tasks, owing to its expansive context window and robust language modeling capabilities.

In contrast, GPT-4, with its advanced in-context learning capabilities, is adept at understanding and generating text based on minimal examples, showcasing significant improvements in NLP tasks without the need for extensive fine-tuning.[7] Moreover, GPT-4 is capable of handling over 25,000 words of text, allowing for use cases like long form content creation, extended conversations, and document search and analysis.[8] In any case, GPT-4 (OpenAI) and Mistral 7B (Mistral) are proprietary models and no specific technical information is available like other open-source models. While for Mistral it is known that there is a version that exploits the *"mixture-of-experts"* technique, which allows optimal performance to be obtained with a much smaller number of trainable parameters, it is not clear whether GPT-4

[7] The Generative Pretrained Transformer (GPT) deviates from the original Transformer by employing a decoder-only architecture, enabling hierarchical processing through its layered structure. This design allows GPT to grasp hierarchical complex semantic patterns in text.

[8] [Online]. Available: https://openai.com/gpt-4

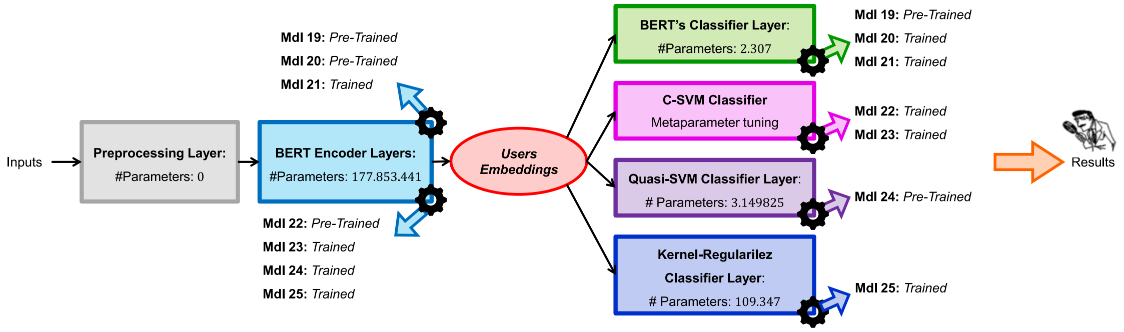


Fig. 3. Training scheme involving BERT from Mdl19 to Mdl25.

uses this specific architecture or is just an improvement of the GPT-3 Transformer.

GPT-4 with 100 trillion parameters, like other suitably capable LLM, has strong capabilities of in-context learning, meaning that it is possible to instruct the model in solving complex tasks by directly feeding the prompt. Hence, in our experiments while Mistral model is used in a standard manner, that is building the embedding vector upon the PPD9 pre-processing, the same pre-processed text is exploited to investigate GPT-4 in-context learning capabilities in Zero-shot setting, designing a suitable procedure (technical details are given in Section V).

To summarize, BERT can be used in various configurations, depending on how many neural network parameters one chooses for the fine-tuning procedure. For these reasons, in our comparative analysis, 7 ways of composing an Embedder-Classifier architecture with the available BERT are proposed. Furthermore, Mistral is a strong contender in generating semantically valid embedding vectors while GPT-4 is investigated in the Zero-shot setting. Hence, the following comparative analysis is carried out:

Mdl19: Complete architecture of pre-trained BERT (Encoder stack + Classification layers) without any parameter training –see Fig. 3;

Mdl20: Complete architecture of pre-trained BERT (Encoder stack + Classification layers) and fine-tuning of only the parameters of the Classification layers –see Fig. 3;

Mdl21: Complete architecture of pre-trained BERT (Encoder stack + Classification layers) and fine-tuning of all the parameters –see Fig. 3. This is the classical usage;

Mdl22: Pre-trained BERT as Embedding Generator (only Encoder stack) without any training. The Classification phase is carried out by an optimized C-SVM –see Fig. 3.

Mdl23: Pre-trained BERT as Embedding Generator (only Encoder stack) and fine-tuning of all the Encoders stack parameters. The Classification phase is carried out by an optimized C-SVM –see Fig. 3.

Mdl24: Pre-trained BERT as Embedding Generator (only Encoder stack) and *Quasi-SVM* neural layer as Classifier and fine-tuning of all the parameters –see Fig. 3.

Mdl25: Pre-trained BERT as Embedding Generator (only Encoder stack) and *Kernel-regularizer* neural layer as Classifier and fine-tuning of all the parameters –see Fig. 3.

Mdl26: Mistral-embed API as embedding generator and an optimized C-SVM for the Classification phase –see Fig. 3.

Mdl27: GPT-4 in a Zero-shot setting, synthesizing a suitable GPTs app for text classification with in-context learning technique.

V. Empirical Results

The entire experimental pipeline was developed using Python. For word-level tokenization, the Python Standard Library was used. The Gensim library [52] was employed for generating N-grams, as it allows for the creation of a customized model capable of generating N-grams from a tokenized corpus, which can be stored and reused on new texts. The implementation of Word2Vec was also taken from the Gensim library. For data cleaning and simple tokenization, the spaCy library[9] was used. Additionally, the Scikit-Learn library [53] was employed for implementing various machine learning routines. Instead, the TensorFlow 2.10.0 library[10] was used for operations with BERT. Finally, for data visualization experiments, the Seaborn [54] library was used.

In the experiments with traditional approaches, models from Mdl1 to Mdl18, and with Hybrid BERT approach, Mdl22 and Mdl23, the classification algorithm adopted is a C-SVM. Its final classification performances are influenced by several hyperparameters that need a careful tuning, notably its kernel (i.e., linear, RBF, polynomial or sigmoidal); the regularization parameter C to drive the maximal-margin optimization problem; the degree of the polynomial (applicable only if the kernel is polynomial); the scale parameter (applicable only if the kernel is polynomial, RBF or sigmoidal).

In our experiments, a 10-fold cross-validation scheme is adopted in order to optimize the SVM hyperparameters with the *balanced accuracy* serving as performance index to validate models in light of the class unbalancing (see Table II). Once the optimal hyperparameters are obtained, the SVM is retrained on $\mathcal{S}_{\text{tr}}$ and then tested on $\mathcal{S}_{\text{ts}}$. Due to the random nature of the k-fold cross-validation procedure, the final performances are obtained by averaging results over five repetitions of the optimization routine.

[9] [Online]. Available: https://spacy.io/

[10] [Online]. Available: https://www.tensorflow.org/

Further settings regarding the word embedding models and BoW-based models can be found in [26]. As instead, in the experiments with Transformer-based architectures, it has been used a pre-trained version of BERT model for the Italian language, called "BERT-Base Multilingual Cased".[11] The pre-trained BERT-Base Multilingual Cased model is composed as follows:

- Encoder stack consisting of $N = 12$ layers;
- Multi-Head Attention layer with $h = 12$ heads;
- Size of the embedding vectors d-model $= 768$;
- A total of 178 M of trainable parameters, as specified in Fig. 3.

When conducting tests that involve the complete BERT architecture, Mdl19-20-21, the classification block is constructed with Linear and SoftMax layers. These layers have the task of predicting a probability distribution of belonging to a class relative to the sequence embeddings. Unlike the Encoder block, these layers have about 2703 trainable parameters and they are not pre-trained, so the parameters are randomly initialized.

Regarding experiments with *hybrid architectures*, the fundamental idea is to leverage both the contextual embeddings produced by BERT's Encoder and the capabilities of SVM for the classification task. Therefore, for models Mdl22 and Mdl23, pre-trained BERT is used to generate embeddings, and *C*-SVM is employed to perform the classification. In this particular instance, as the architecture comprises algorithms with distinct conceptual training approaches, it was essential to conduct separate training for each block.

Additional experiments are suggested using a hybrid architecture, Mdl24 and Mdl25, that maintains the use of pre-trained BERT for embedding generation. However, rather than employing a conventional *C*-SVM classifier, a neural approximation of *C*-SVM is introduced. This facilitates the simultaneous training of both the Embedder and Classifier components. To carry out these experiments, the previously described pre-trained BERT is used, to which is added one of these Classification blocks:

- Quasi-SVM block, composed of 3.149.825 non-trainable parameters;
- Kernel regularizer block, composed of 109.347 trainable parameters.

As the last two models are concerned, Mdl26 leverages the Mistral Embeddings API[12] to query Mistral 7B in order to retrieve the authors embedding vectors whose dimension is 1024.

Finally, for GPT-4, we built an instance of an *in-context learning text classifier*, by developing a GPTs. GPTs is a kind of plug-in of GPT-4 that can be developed by any user; all that's needed is to provide a detailed description of the task to be performed and then explain to the model (GPT-4) how it should behave. The description used to build the *In-Context Classifier* GPTs contains the main concept of the in-context learning and the execution pipeline. The execution pipeline involves:

- Ingesting the dataset file, detailing its structure to guide GPT-4 on the analysis material.
- Implementing batch analysis, sequentially proceeding with a user-initiated 'go' prompt.
- Conducting batch analysis using GPT-4's in-context classification capabilities.
- Outputting results in JSON format, detailing the expected structure for clarity.
- Tracking and displaying analysis progress, including partial classification results in JSON format.

During the use phase of the *In-Context Classifier*, we prepare GPT-4 by supplying the dataset alongside a comprehensive explanation of labels. Label descriptions may be periodically updated to enhance understanding. In instances of uncertainty, GPT-4 is prompted to engage in deeper reasoning about the task at hand, ensuring a thorough analysis. Finally, the GPTs output related to the in-context classification is saved and further processed to summarize the results. GPT-4, as expected, demonstrated high semantic capabilities, providing also explanations when it failed to assign the class label because the post was off-topic.

A. Classification Results

The classification step offers a measure of the generalization ability of the models under analysis. In particular, as performance indices to evaluate the model, starting from the confusion matrix, we consider: precision (macro, micro, weighted), recall (macro, micro, weighted), balanced accuracy, accuracy and F_1 (macro, micro, weighted). The performance results, obtained by testing all the models described in Sections IV-B, IV-C and IV-E, are reported in Table III. Therefore, three characteristics –measured on the same $\mathcal{S}_{\mathrm{ts}}$ for each of the 27 considered models to ensure a fair comparison– are taken into account in the overall performance evaluation and discussion. The first concerns the ability of the tested models to address the problem of unbalanced classes through balanced accuracy (average between sensitivity and specificity), the second concerns the F_1 score (harmonic mean of precision and recall), which gives a balanced measure of the classifier's ability to correctly classify both positive and negative instances. The third feature concerns the semantic capabilities of the embedding models and will be separately discussed in Section V-B. On the one hand, the discussion of the results will help us choose the best methodology in terms of application, but on the other it will help us to ground some general considerations, with a scientific flavor, on the semantic capabilities of the investigated models.

The best results, with a balanced accuracy of 99.4%, has been obtained through Mistral embedding (Mdl26). By misclassifying just one item, Mistral demonstrates enormous semantic capacity even in the Italian language. Mdl27, which leverages GPT-4 for in-context learning LLM-based classification, demonstrates adaptability and efficiency in text classification, particularly noteworthy given the niche application and the usage of the Italian language. Its performance can be closely compared to other models that combine traditional NLP approaches with deep learning techniques. Performances are

[11] [Online]. Available: https://github.com/google-research/bert/blob/master/multilingual.md

[12] [Online]. Available: https://docs.mistral.ai/guides/embeddings/

TABLE III
COMPARISON AMONGST ALL OF THE 25 TESTED MODELS

	Precision macro	Precision micro	Precision weighted	Recall micro	Recall macro	Recall weighted	Balanced accuracy	Accuracy	F_1 macro	F_1 micro	F_1 weighted
Language models trained from scratch											
Mdl1	0.772 ± 0.015	0.851 ± 0.013	0.849 ± 0.007	0.851 ± 0.013	0.743 ± 0.026	0.851 ± 0.013	**0.743 ± 0.026**	0.851 ± 0.013	0.753 ± 0.013	0.851 ± 0.013	0.848 ± 0.010
Mdl2	0.729 ± 0.057	0.814 ± 0.026	0.807 ± 0.029	0.814 ± 0.026	0.677 ± 0.030	0.814 ± 0.026	0.677 ± 0.030	0.814 ± 0.026	0.696 ± 0.038	0.814 ± 0.026	0.808 ± 0.026
Mdl3	0.767 ± 0.020	0.847 ± 0.011	0.840 ± 0.013	0.847 ± 0.011	0.716 ± 0.018	0.847 ± 0.011	0.716 ± 0.018	0.847 ± 0.011	0.735 ± 0.019	0.847 ± 0.011	0.841 ± 0.012
Mdl4	0.790 ± 0.017	0.863 ± 0.012	0.859 ± 0.010	0.863 ± 0.012	0.742 ± 0.012	0.863 ± 0.012	0.742 ± 0.012	0.863 ± 0.012	0.762 ± 0.010	0.863 ± 0.012	0.859 ± 0.010
Pretrained Language Models											
Mdl5	0.910 ± 0.037	0.933 ± 0.029	0.936 ± 0.028	0.933 ± 0.029	0.882 ± 0.048	0.933 ± 0.029	**0.882 ± 0.048**	0.933 ± 0.029	0.894 ± 0.043	0.933 ± 0.029	0.933 ± 0.030
Mdl6	0.848 ± 0.016	0.885 ± 0.006	0.885 ± 0.005	0.885 ± 0.006	0.744 ± 0.014	0.885 ± 0.006	0.744 ± 0.014	0.885 ± 0.006	0.780 ± 0.009	0.885 ± 0.006	0.878 ± 0.003
Language models trained from scratch + data cleaning											
Mdl7	0.900 ± 0.082	0.901 ± 0.009	0.906 ± 0.011	0.901 ± 0.009	0.717 ± 0.029	0.901 ± 0.009	0.717 ± 0.029	0.901 ± 0.009	0.732 ± 0.030	0.900 ± 0.009	0.885 ± 0.010
Mdl8	0.694 ± 0.015	0.845 ± 0.015	0.844 ± 0.011	0.845 ± 0.015	0.682 ± 0.012	0.845 ± 0.015	0.682 ± 0.012	0.845 ± 0.015	0.687 ± 0.013	0.845 ± 0.015	0.844 ± 0.013
Mdl9	0.879 ± 0.061	0.919 ± 0.007	0.916 ± 0.009	0.919 ± 0.007	0.770 ± 0.030	0.919 ± 0.007	**0.770 ± 0.030**	0.919 ± 0.007	0.799 ± 0.035	0.919 ± 0.007	0.911 ± 0.011
Mdl10	0.731 ± 0.024	0.877 ± 0.009	0.868 ± 0.006	0.877 ± 0.009	0.717 ± 0.015	0.877 ± 0.009	0.717 ± 0.015	0.877 ± 0.009	0.722 ± 0.019	0.877 ± 0.009	0.872 ± 0.007
Hybrid language models based on concatenation of different embeddings											
Mdl11	0.778 ± 0.031	0.857 ± 0.013	0.853 ± 0.019	0.857 ± 0.013	0.729 ± 0.023	0.857 ± 0.013	0.729 ± 0.023	0.857 ± 0.013	0.749 ± 0.026	0.857 ± 0.013	0.852 ± 0.015
Mdl12	0.768 ± 0.025	0.853 ± 0.014	0.850 ± 0.016	0.853 ± 0.014	0.725 ± 0.025	0.853 ± 0.014	0.725 ± 0.025	0.853 ± 0.014	0.743 ± 0.025	0.853 ± 0.014	0.849 ± 0.015
Mdl13	0.824 ± 0.046	0.873 ± 0.021	0.875 ± 0.023	0.873 ± 0.021	0.763 ± 0.024	0.873 ± 0.021	**0.763 ± 0.024**	0.873 ± 0.021	0.788 ± 0.032	0.873 ± 0.021	0.869 ± 0.020
Mdl14	0.767 ± 0.029	0.851 ± 0.013	0.848 ± 0.012	0.851 ± 0.013	0.738 ± 0.019	0.851 ± 0.013	0.738 ± 0.019	0.851 ± 0.013	0.751 ± 0.020	0.851 ± 0.013	0.848 ± 0.012
Traditional Approaches based on BoW–TF-IDF–LSA											
Mdl15	0.276 ± 0.046	0.533 ± 0.025	0.429 ± 0.045	0.533 ± 0.025	0.317 ± 0.005	0.533 ± 0.025	0.317 ± 0.005	0.533 ± 0.025	0.284 ± 0.023	0.533 ± 0.025	0.463 ± 0.016
Mdl16	0.305 ± 0.005	0.531 ± 0.006	0.457 ± 0.005	0.531 ± 0.006	0.321 ± 0.003	0.531 ± 0.006	0.321 ± 0.003	0.531 ± 0.006	0.298 ± 0.003	0.531 ± 0.006	0.475 ± 0.004
Mdl17	0.302 ± 0.008	0.527 ± 0.009	0.455 ± 0.008	0.527 ± 0.009	0.321 ± 0.005	0.527 ± 0.009	**0.321 ± 0.005**	0.527 ± 0.009	0.300 ± 0.006	0.527 ± 0.009	0.476 ± 0.007
Mdl18	0.282 ± 0.048	0.537 ± 0.023	0.435 ± 0.048	0.537 ± 0.023	0.321 ± 0.003	0.537 ± 0.023	0.321 ± 0.003	0.537 ± 0.023	0.287 ± 0.025	0.537 ± 0.023	0.467 ± 0.018
Pure Transformer-based Architecture: BERT											
Mdl19	0.100 ± 0.074	0.269 ± 0.212	0.117 ± 0.143	0.269 ± 0.212	0.327 ± 0.020	0.269 ± 0.212	0.327 ± 0.020	0.269 ± 0.212	0.140 ± 0.089	0.269 ± 0.212	0.157 ± 0.177
Mdl20	0.749 ± 0.074	0.705 ± 0.005	0.713 ± 0.010	0.705 ± 0.005	0.558 ± 0.000	0.705 ± 0.005	0.558 ± 0.000	0.705 ± 0.005	0.594 ± 0.0045	0.705 ± 0.005	0.692 ± 0.003
Mdl21	0.940 ± 0.030	0.962 ± 0.009	0.962 ± 0.008	0.962 ± 0.009	0.921 ± 0.020	0.962 ± 0.009	**0.921 ± 0.020**	0.962 ± 0.009	0.930 ± 0.021	0.962 ± 0.009	0.961 ± 0.008
Hybrid Architecture: BERT and SVM Model											
Mdl22	0.569 ± 0.000	0.677 ± 0.000	0.667 ± 0.000	0.677 ± 0.000	0.541 ± 0.000	0.677 ± 0.000	0.541 ± 0.000	0.677 ± 0.000	0.552 ± 0.000	0.677 ± 0.000	0.670 ± 0.000
Mdl23	0.914 ± 0.034	0.943 ± 0.012	0.944 ± 0.013	0.943 ± 0.012	0.888 ± 0.032	0.943 ± 0.012	**0.888 ± 0.032**	0.943 ± 0.012	0.897 ± 0.017	0.943 ± 0.012	0.943 ± 0.011
Hybrid Architecture: BERT and Neural SVM Model											
Mdl24	0.304 ± 0.039	0.461 ± 0.046	0.445 ± 0.042	0.461 ± 0.046	0.308 ± 0.035	0.461 ± 0.046	0.308 ± 0.035	0.461 ± 0.046	0.305 ± 0.036	0.461 ± 0.046	0.452 ± 0.043
Mdl25	0.293 ± 0.066	0.440 ± 0.051	0.429 ± 0.049	0.440 ± 0.051	0.293 ± 0.052	0.440 ± 0.051	0.293 ± 0.052	0.440 ± 0.0513	0.292 ± 0.056	0.440 ± 0.051	0.434 ± 0.049
Pretrained Multilingual Large Language Model (Mistral)											
Mdl26	0.963 ± 0.000	0.999 ± 0.000	0.991 ± 0.000	0.999 ± 0.000	0.994 ± 0.000	0.999 ± 0.000	**0.994 ± 0.000**	0.999 ± 0.000	0.978 ± 0.000	0.999 ± 0.000	0.990 ± 0.000
In-context learning Classifier with GPT-4											
Mdl27	0.717 ± 0.187	0.798 ± 0.151	0.802 ± 0.153	0.798 ± 0.151	0.723 ± 0.187	0.798 ± 0.151	**0.723 ± 0.187**	0.798 ± 0.151	0.718 ± 0.185	0.798 ± 0.151	0.799 ± 0.152

Results include 11 performance indices evaluated on the test set in terms of average ± standard deviation across 5 different runs of the training and testing procedure. Results for Mdl1 to Mdl18 are quoted from [26].

comparable to models like Mdl1 to Mdl7 or Mdl11, which range from language models trained from scratch to hybrid embedding with heavy data cleaning pipelines. Second best results are achieved by the BERT architecture (Mdl21) used in a classical way by operating the fine-tuning of all the parameters. The Balanced Accuracy reaches 92% with a 96% weighted F_1 score. The poor results of the other two models (Mdl19 and Mdl20) are expected and reported for the sake of comparison. The second group containing the best performing models is that relating to the use of BERT as a feature extractor (with fine tuning) and the C-SVM as a classifier (Mdl23) with a Balanced Accuracy of 88.8%. The other case (Mdl22) in which no training of the BERT parameters was carried out proved to be really poor performing, as expected. In light of these results, we can say that the performances of Mdl23 are comparable with those of Mdl5 (weighted Balanced Accuracy and F_1 score are 88.2% and 93%, respectively). This means that the generalization capability of BERT+C-SVM (Mdl23) is similar to one obtained using a pre-trained word embedding model. This is interesting because, at least on this dataset, word embedding models perform well and could be a great alternative to BERT (considering the trade-off between accuracy and computational burden). Interestingly, the language models trained from scratch achieve good performances (slightly lower than the previous case) if a careful data cleaning stage (Mdl9) is performed concerning the base case (Mdl1-2-3-4) demonstrating, as known, that such models are sensitive to noise. Also, the concatenation of user vectors obtained with different embedding models (Hybrid language models) achieves good performances (weighted Balanced Accuracy and F_1 score are 76% and 89%, respectively), especially if the embedding model trained from scratch is based on N-grams (PPD4 – see Section IV-A). Hybrid models demonstrate discrete generalization capabilities. From an information-theoretic perspective, combining embedding vectors from two distinct language models into a single vector tends to enhance the information content within the embedding. However, this increase in information comes at the cost of longer user vectors, posing challenges during the classification phase due to the familiar curse of dimensionality. In contrast, when employing traditional models like the BoW–TF-IDF–LSA pipeline, the results are notably poor. No specific model outperforms the others, suggesting a classification that resembles random chance. These models completely falter in accurately representing users for our specific problem.

Bad results are similarly obtained with the BERT architecture hybridized with a Neural SVM model. Analyzing the results, we hypothesize that in this case the low performance is due to the non-optimizated parameters of the Classification block. Specifically, for Mdl24, the pre-trained *RandomFourierFeatures* layer is used [51], which does not allow further fine-tuning. On the other hand, for Mdl25, the number, type, and activation functions of the intermediate layers in the Classification block have not been optimized. In fact, the approximation between SVM and Neural SVM imposes constraints on the loss function and regularization factor, but not on other metaparameters of the neural network.

Therefore considering the results in general, from an application point of view, it is interesting to note that word embedding models can be a valid alternative, especially in hardware environments with limited computational resources. The important thing, however, is to find the right pre-processing procedure, specifically in terms of data cleaning. BERT in its classic use is the choice of choice above all because it allows to omit the data cleaning phase altogether, or in any case to reduce it to a

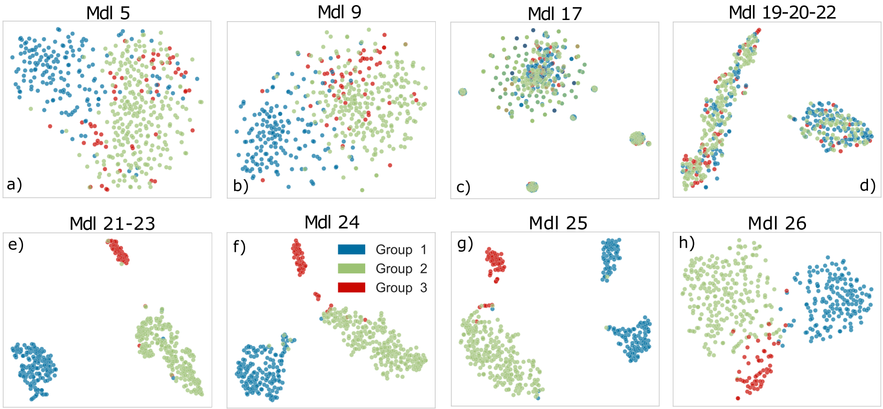


Fig. 4. t-SNE visualization of the user-vector space representation for several embedding/pre-processing models, including BERT experiments – panels (d), (e), (f), (g). Panels (a), (b), (c) are taken from our previous work [26]. Colors represent the original class labels. Groups (i.e., classes) are ordered according to Table II.

minimum. Similarly, BERT proves to be optimal in the semantic representation of the language since, as we know, it incorporates the power of word embedding models and adds contextual embedding as well as information relating to the ordering of tokens. The other methodologies that we have experimented as BERT hybridizations, at least for the examined dataset, have not provided interesting results and therefore it will not be possible to exploit their main characteristics which are the low computational cost.

Both the experiments with LLMs show interesting results from an application and scientific point of view, thanks to the exceptional generalization degree reached through Mistral embedding and the very good text understanding and classification capability demonstrated by GPT-4. Moreover, as mentioned in Section IV-E, this methodology can provide semantically relevant explanations regarding the classification and discard irrelevant texts according to the chosen classes, also providing explanations in this case. In fact, Mdl27's unique capability to perform in-context learning sets it apart, highlighting the advanced progression in LLMs towards more intuitive, context-aware processing in multilingual settings. This model exemplifies the growing trend toward leveraging deep learning models' nuanced understanding of language for complex classification tasks. As a final note on GPT-4, we point out that, although this model has demonstrated good generalization ability, the current model (at the time of writing) appears to be affected by laziness, a problem also recognized by OpenAI. A very interesting point, however, is the ability to not assign a label to posts that are actually off-topic, such as the following "Scusate la domanda, non sarebbe meglio che il gruppo fosse chiuso anche per un fatto di privacy?".[13] This shows the power of this language model which, we reiterate, thanks to in-context learning, acts at a higher semantic level than techniques based on embedding.

B. Assessing the Semantic Capabilities of Models

To visualize and explore how each representation captures the underlying semantic relationships within texts, we created a 2D scatter plot using t-SNE (t-distributed Stochastic Neighbor Embedding) [55]. Fig. 4 displays these scatter plots, where each point represents a user-vector, and different colors distinguish different classes. The user-vectors are obtained from the best-performing model in each major embedding approach. This visualization allows us to observe how well the embeddings separate and cluster users based on their classes. As regards the plots of the user-vectors projected in the reduced dimension space, also in relation to the results relating to the classification performance, we can make two orders of observations. As regards the vectors obtained with BERT (panels d), e), f), g) of Fig. 4) in all cases drawn that for Mdl19, Mdl20 and Mdl21 (panel d)) the vectors show an excellent tendency to cluster in a manner congruent to the class to which they belong. Same observation can be made for Mistral (Mdl26), that reach optima classification performances (panel h)). As regards Mdl21, Mdl23 (panel e)) and Mdl24 (panel f)) in all cases the clusters are very compact, specifically for class 3 (Group 3). However, we know that the first two models achieve high performance in terms of accuracy, while Mdl24 performs poorly. In panels a) and b) of Fig. 4, we can observe that the neural-based language models (Mdl5 and Mdl9), exhibit superior grouping capabilities for user embedding vectors, indicating more effective class discrimination. Conversely, in the case of models relying on the BoW, TF-IDF, and LSA pipeline (as shown in panel c) of Fig. 4), there is a noticeable deficiency in clustering ability. This leads to significant overlap among users from different classes, thereby complicating the classification task. The t-SNE plots further reveal an overlap between two specific classes (red and green), even for the most proficient models. However, this issue is absent when employing BERT, which, as widely recognized, excels in generating contextual embeddings. This disparity can be rationalized by considering that the affected classes primarily comprise users discussing cancer in various forms. BERT's representation manages to capture the similarities between these users more effectively compared to all other cases.

[13] Transl. "Sorry for the question, wouldn't it be better if the group were closed also for reasons of privacy?"

TABLE IV
SENTENCES IN ITALIAN WITH THEIR RESPECTIVE TRANSLATION AND SEGMENTATION THROUGH BERT'S WORDPIECE TOKENIZATION

Group 1: Dieta e diabete di tipo 2
Soffro di diabete mellito. Lo conoscete? [Transl:*I suffer from diabetes mellitus. Do you know what is it?* `Start So ##ff ##ro di dia ##bet ##e me ##lli ##to . Lo conoscete ? Stop`
Il diabete autoimmune è dovuto a carenze ormonali?? [Transl:*Is autoimmune diabetes due to hormone deficiencies??* `Start Il dia ##bet ##e auto ##im ##mune è dovuto a care ##nze or ##mona ##li ? ? Stop`
Group 2: Tumore al colon-retto, restiamo vicini
Buonasera, che sintomi avevate prima di scoprire il tumore al colon? [Transl:*Good evening, what symptoms did you have before discovering colon cancer?* `Start Bu ##onas ##era , che sin ##tomi aveva ##te prima di s ##co ##prire il tumor ##e al col ##on ? Stop`
Come gestite la nausea dovuta ai liquidi da bere per la colonscopia? [Transl:*How do you handle nausea from drinking colonoscopy fluids?* `Start Come ge ##sti ##te la nau ##sea dovuta ai liquid ##i da bere per la colons ##co ##pia ? Stop`
Group 3: Italia-glioblastoma multiforme-cancro al cervello
ho scoperto da poco che mia mamma ha un GBM. ho paura. [Transl:*I recently found out that my mom has GBM. I am afraid.* `Start ho scoperto da poco che mia mam ##ma ha un GB ##M . ho paura . Stop`
Sapete del dicloracetato di sodio nella cura del GBM? [Transl:*Do you know about sodium dichloroacetate in the treatment of GBM?* `Start Sa ##pet ##e del di ##clo ##race ##tato di so ##dio nella cura del GB ##M ? Stop`

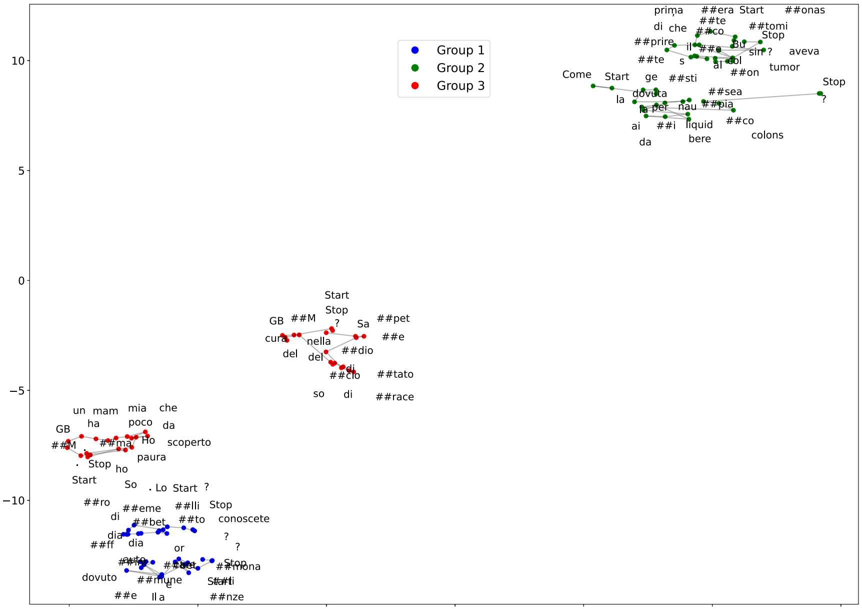


Fig. 5. t-SNE visualization of word contextual embeddings, obtained with BERT (Mdl21), of the tokens for 9 selected sentences – see Table IV.

In summary, various experiments have been conducted to investigate the distinctive features and exceptional capabilities of BERT's Contextual Embeddings. Typically, for practical applications, BERT's Sentence Embeddings are predominantly utilized, representing a refined high-level form of word embeddings for the tokens within a sentence. As explained in Section IV-E, each token's embedding vector is unique, thanks to the three encodings applied during pre-processing. This characteristic makes the methodology particularly effective in addressing challenges related to *coreference* and *polysemy*. The present analysis aims to provide a detailed visualization of the word embeddings related to 9 sentences – three for each group – extracted from the Facebook users' dataset. These sentences are listed in Table IV, highlighting their tokenizations applied by BERT's WordPiece tokenizer.[14] Fig. 5 displays the representation of BERT's Contextual Embeddings – the classical BERT adopted for Mdl21 – for the tokens of the selected sentences, obtained with t-SNE approach. For the sake of interpretation, links between consecutive tokens within the same sentence are also reported. From Fig. 5, it is evident that there is a significant separation of word embeddings, both between different groups and among sentences within the same group, except for two sentences in Group 1, where there is some overlap due to both discussing "colonoscopy". Furthermore, we can also observe the presence of distinct word embeddings for the same token – for example `colons` – which confirms the uniqueness of Contextual Embeddings attributed to the essential Positional Encoder.This brief example shows the great ability of BERT to represent the various linguistic aspects of a text in a multidimensional way and this justifies the high performance, also measured in the dataset under analysis, both as a feature extractor and as a textual classifier.

C. Implications of the Study

The study highlights the use of LLMs and embedding techniques for analyzing medical discussions on platforms like

[14] [Online]. Available: https://pypi.org/project/word-piece-tokenizer/

Facebook. It demonstrates how these models identify complex patterns and sentiments in health discourse, consisting in noisy and short texts in Italian (a language often underused in language models), offering insights into public health trends with the potential of fighting misinformation (a paramount issue especially when health is at stake) by casting a suitable classification problem. The implication of the following study is significant, showcasing a scalable method for healthcare professionals to understand patient experiences, leading to better public health strategies. Moreover, modern LLM further advance this by inherently grasping context and semantics without extensive preprocessing, streamlining the analysis of natural language data. These developments mark a significant shift towards more intuitive and efficient text processing methodologies, enhancing the ability to extract meaningful insights from vast textual datasets. Interestingly, GPT-4's in-context learning ability, as emerging methodology, introduces a refined approach to text classification, minimizing the need for large datasets and enabling nuanced analysis opening the way to truly Explainable AI systems.

VI. Conclusion

This investigation deals with the challenging problem of classifying users discussing medical topics on a generalist SN, where the posts are short, noisy and, in our experiments, written in Italian language. Different semantic text representation strategies and pre-processing pipelines were compared in order to establish which strategies achieve optimal performance in terms of generalization capabilities. From a methodological point of view, various embedding techniques have been tested, from the traditional ones up to state-of-the-art techniques such as BERT, in some variants, Mistral and GPT-4. In this way, the logic behind the experiments has re-proposed, albeit synthetically and without claiming completeness, the historical path that has seen a drastic improvement in performance with the advent of the use of LLMs with Transformer architectures in the realm of text mining and NLP. The study demonstrates that although Mistral and BERT are optimal in classifying short and noisy texts, word embedding techniques can still be a viable alternative, but need accurate data pre-processing. We also experimented with the possibility of training word embedding models on a huge dataset of leaflets to verify the semantic representation capability of trained models. Also in this case we obtained interesting performances. As far as the classification procedure itself is concerned, apart from the classic BERT pipeline which foresees dense layers as a classifier, we have found that hybrid techniques proposed in the literature between BERT and SVM (also in the case known as Quasi-SVM in which SVM is approximated with a neural architecture) do not provide good results, at least for the dataset at hand. Ultimately, unless constrained by strong computational limitations, Mistral 7B is the optimal choice as an algorithm within a decision support system that can monitor users on specific and sensitive discourses such as those in the medical field, where fake news and infodemic can have a dramatic impact. Conversely, in the presence of strong computational limitations, word embedding techniques can be safely used, even for short and noisy posts with medical content in the Italian language. GPT-4 also demonstrates strong in-context learning capabilities, particularly in the Zero-shot setting, taking advantage of the possibility of using the model as a text classifier for medical discussions. Future work involves an in-depth study of the semantic representation and embedding capabilities of LLMs and the document classification capabilities by exploiting the in-context learning technique for GPT-4 and other open-source LLMs also in the Few-shot setting.

Acknowledgment

The authors would like to thank association Arianna ODV for having contributed to the work and having provided indications on patients needs.

References

[1] R. Kurzweil, "The singularity is near," in *Ethics and Emerging Technologies*. Berlin, Germany: Springer, 2005, pp. 393–406.

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.

[3] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11, vol. 30.

[4] E. De Santis, G. De Santis, and A. Rizzi, "Multifractal characterization of texts for pattern recognition: On the complexity of morphological structures in modern and ancient languages," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 10143–10160, Aug. 2023.

[5] E. De Santis, A. Martino, and A. Rizzi, "Human versus machine intelligence: Assessing natural language generation models through complex systems theory," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 7, pp. 4812–4829, Jul. 2024.

[6] T. Stanisz, S. Drożdż, and J. Kwapień, "Complex systems approach to natural language," *Phys. Rep.*, vol. 1053, pp. 1–84, 2024.

[7] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, vol. 13.

[8] T. Mikolov, W.-T. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2013, pp. 746–751.

[9] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[10] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Amer. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.

[11] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2177–2185.

[12] K. Lund and C. Burgess, "Producing high-dimensional semantic spaces from lexical co-occurrence," *Behav. Res. Methods, Instrum., Comput.*, vol. 28, pp. 203–208, 1996.

[13] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, 2017.

[14] N. Malcolm, "Wittgenstein's philosophical investigations," *Philos. Rev.*, vol. 63, no. 4, pp. 530–559, 1954.

[15] E. De Santis and A. Rizzi, "Prototype theory meets word embedding: A novel approach for text categorization via granular computing," *Cogn. Comput.*, vol. 15, no. 3, pp. 976–997, 2023.

[16] H. Lee, "The rise of chatGPT: Exploring its potential in medical education," *Anat. Sci. Educ.*, vol. 17, no. 5, pp. 926–31, 2024, doi: 10.1002/ase.2270.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, vol. 1, 2019, pp. 4171–4186.

[18] T. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 1877–1901.

[19] J. Achiam et al., "GPT-4 technical report," 2023, *arXiv:2303.08774*.

[20] A. Q. Jiang et al., "Mistral 7b," 2023, *arXiv:2310.06825*.

[21] Q. Dong et al., "A survey for in-context learning," 2022, *arXiv:2301.00234*.

[22] G. Gonzalez-Hernandez, A. Sarker, K. O'Connor, and G. Savova, "Capturing the patient's perspective: A review of advances in natural language processing of health-related text," *Yearbook Med. Inform.*, vol. 26, no. 01, pp. 214–227, 2017.

[23] E. De Santis, A. Martino, and A. Rizzi, "An infoveillance system for detecting and tracking relevant topics from italian tweets during the COVID-19 event," *IEEE Access*, vol. 8, pp. 132527–132538, 2020.

[24] E. De Santis, A. Martino, F. Ronci, and A. Rizzi, "An unsupervised graph-based approach for detecting relevant topics: A case study on the italian twitter cohort during the Russia- Ukraine conflict," *Information*, vol. 14, no. 6, 2023, Art. no. 330. [Online]. Available: https://www.mdpi.com/2078-2489/14/6/330

[25] S. B. Naeem, R. Bhatti, and A. Khan, "An exploration of how fake news is taking over social media and putting public health at risk," *Health Inf. Libraries J.*, vol. 38, no. 2, pp. 143–149, 2021.

[26] E. De Santis, A. Martino, F. Ronci, and A. Rizzi, "A comparison of neural word embedding language models for classifying social media users in the healthcare context," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2023, pp. 1–9.

[27] Y. Tang, "Deep learning using support vector machines," 2013, *arXiv:1306.0239*.

[28] M. Jändel, "A neural support vector machine," *Neural Netw.*, vol. 23, no. 5, pp. 607–613, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608010000043

[29] D. Demner-Fushman, W. W. Chapman, and C. J. McDonald, "What can natural language processing do for clinical decision support?," *J. Biomed. Inform.*, vol. 42, no. 5, pp. 760–772, 2009.

[30] T. Ching et al., "Opportunities and obstacles for deep learning in biology and medicine," *J. Roy. Soc. Interface*, vol. 15, no. 141, 2018, Art. no. 20170387.

[31] A. Canosa et al., "Brain metabolic differences between pure bulbar and pure spinal als: A 2-[18f]fdg-pet study," *J. Neurol.*, vol. 270, no. 2, pp. 953–959, 2023, doi: 10.1007/s00415-022-11445-9.

[32] I. Alimova and E. Tutubalina, "Automated detection of adverse drug reactions from social media posts with machine learning," in *Analysis of Images, Social Networks and Texts*, W. M. van der Aalst et al., Eds. Cham, Switzerland: Springer, 2018, pp. 3–15.

[33] Y. Zhang, D. He, and Y. Sang, "Facebook as a platform for health information and communication: A case study of a diabetes group," *J. Med. Syst.*, vol. 37, no. 9942, pp. 1–12, 2013.

[34] B. Chiu, G. Crichton, A. Korhonen, and S. Pyysalo, "How to train good word embeddings for biomedical NLP," in *Proc. 15th Workshop Biomed. Natural Lang. Process.*, 2016, pp. 166–174.

[35] Y. Wang et al., "A comparison of word embeddings for the biomedical natural language processing," *J. Biomed. Inform.*, vol. 87, pp. 12–20, 2018.

[36] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.

[37] X. Wang, M. Tao, R. Wang, and L. Zhang, "Reduce the medical burden: An automatic medical triage system using text classification bert based on transformer structure," in *Proc. IEEE 2nd Int. Conf. Big Data Artif. Intell. Softw. Eng.*, 2021, pp. 679–685.

[38] T. Wang, K. Lu, K. P. Chow, and Q. Zhu, "COVID-19 sensing: Negative sentiment analysis on social media in China via bert model," *IEEE Access*, vol. 8, pp. 138162–138169, 2020.

[39] R. Qasim et al., "A fine-tuned bert-based transfer learning approach for text classification," *J. Healthcare Eng.*, vol. 2022, no. 1, 2022, Art. no. 3498123.

[40] A. L. Tonja et al., "CIC NLP at SMM4h 2022: A BERT-based approach for classification of social media forum posts," in *Proc. 7th Workshop Social Media Mining Health Appl., Workshop Shared Task*, 2022, pp. 58–61.

[41] Y. Liu et al., "RoBERTa: A robustly optimized bert pretraining approach," 2019, *arXiv:1907.11692*.

[42] A. Murarka, B. Radhakrishnan, and S. Ravichandran, "Classification of mental illnesses on social media using RoBERTa," in *Proc. 12th Int. Workshop Health Text Mining Inf. Anal.*, 2021, pp. 59–68.

[43] N. A. Asad, P. Mahmud, Md. A. Afreen, S. Islam, and Md. Maynul, "Depression detection by analyzing social media posts of user," in *Proc. IEEE Int. Conf. Signal Process., Inf., Commun. & Syst.*, 2019, pp. 13–17, doi: 10.1109/SPICSCON48833.2019.9065101.

[44] C. Zhang and H. Yamana, "WUY at SemEval-2020 task 7: Combining BERT and naive Bayes-SVM for humor assessment in edited news headlines," in *Proc. 14th Workshop Semantic Eval. Barcelona (Online): Int. Committee Comput. Linguistics*, 2020, pp. 1071–1076. [Online]. Available: https://aclanthology.org/2020.semeval-1.141

[45] Y. Chen, W. Huang, L. Nguyen, and T.-W. Weng, "On the equivalence between neural network and support vector machine," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 23478–23490.

[46] S. Kang and S. Cho, "Approximating support vector machine with artificial neural network for fast prediction," *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4989–4995, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417414000888

[47] N. Muennighoff et al., "Generative representational instruction tuning," in *Proc. Int. Conf. Learn. Representations Workshop: How Far Are We From AGI*, 2024.

[48] Y. Mathur, S. Rangreji, R. Kapoor, M. Palavalli, A. Bertsch, and M. R. Gormley, "Summqa at mediqa-chat 2023: In-context learning with gpt-4 for medical summarization," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2023, pp. 490–502.

[49] A. Martino, E. De Santis, and A. Rizzi, "An ecology-based index for text embedding and classification," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2020, pp. 1–8.

[50] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[51] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., vol. 20, pp. 1–8, 2007. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf

[52] R. Řehuřek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proc. LREC Workshop New Challenges NLP Frameworks*. Valletta, Malta, 2010, pp. 45–50.

[53] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

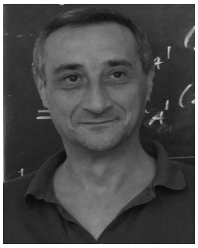
[54] M. L. Waskom, "seaborn: Statistical data visualization," *J. Open Source Softw.*, vol. 6, no. 60, 2021, Art. no. 3021, doi: 10.21105/joss.03021.

[55] L. Van Der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

Enrico De Santis (Member, IEEE) received the M.A.Sc. (Hons.) and the Ph.D. degrees in information and communication engineering from the "Sapienza" University of Rome, Rome, Italy. He was an Assistant Researcher and a Postdoc with the Department of Computer Science, Toronto Metropolitan University, Toronto, ON, Canada. He is currently a Researcher with the Department of Information Engineering, Electronics and Telecommunications, "Sapienza". In 2017, he has joined an innovative startup with "Sapienza" University as CTO, dealing with the management of Artificial Intelligence projects in production environments. His research interests include artificial intelligence, complex systems and data-driven modeling, natural language processing, computational intelligence, neural networks, and fuzzy systems with application in smart grids and predictive maintenance.

Alessio Martino (Member, IEEE) received the graduation (*summa cum laude*) degree in communications engineering from the University of Rome "La Sapienza", Rome, Italy, in 2016. His bachelor's and master's degrees Theses regarded EU-FP7 and EU-FP8 projects, respectively. From 2016 to 2019, he was a Ph.D. Research Fellow of information and communications technologies with the Department of Information Engineering, Electronics and Telecommunications, University of Rome "La Sapienza" with a final dissertation on pattern recognition techniques in non-metric domains. During his Ph.D., he was a Scientific Collaborator with Consortium for Research in Automation and Telecommunication, Rome. After his Ph.D. degree, he has been granted a 1-year PostDoctoral Research Fellowship with the University of Rome "La Sapienza" and a 1-year PostDoctoral Research Fellowship with the Italian National Research Council. He is currently an Assistant Professor of computer science with LUISS University, Rome. His research interests include machine learning, computational intelligence and knowledge discovery. He is focusing on large-scale machine learning, advanced pattern recognition systems, Big Data analysis, parallel and distributed computing, granular computing, and complex systems modeling, in applications including bioinformatics and computational biology, natural language processing, and energy distribution networks.

Francesca Ronci received the M.A.Sc. degree in electronic engineering from the "La Sapienza" University of Rome, Rome, Italy. In particular, he has specialized her academic training by taking an interest in AI and ML, focusing on NLP using both neural networks and traditional methodologies, with application to several technical areas, such as sentiment analysis, social network analysis, and automatic text generation. In 2021, she has joined the innovative startup Sis.Ter.Pomos with "La Sapienza" University as a consultant, dealing with AI projects.



Antonello Rizzi (Senior Member, IEEE) has been with the Department of Information Engineering, Electronics and Telecommunications (DIET), "Sapienza" University of Rome, Rome, Italy, as an Assistant Professor, since July 2010. He is currently an Associate Professor with DIET. Since 2008, he has been the scientific coordinator and R&D technical Director with the Intelligent Systems Laboratory within the Research Center for Sustainable Mobility of Lazio region, Italy. His main research interests include computational intelligence and pattern recognition, supervised and unsupervised machine learning techniques, neural networks, fuzzy systems, and evolutionary algorithms, with application in smart grids and microgrids modeling and control, intelligent systems for sustainable mobility, and battery management systems. He has coauthored more than 220 international journal/conference papers and book chapters.