PLOS ONE

RESEARCH ARTICLE

A survey on text classification: Practical perspectives on the Italian language

Andrea Gasparetto[1]*, **Alessandro Zangari**[1]☯, **Matteo Marcuzzo**[1]☯, **Andrea Albarelli**[2]

1 Department of Management, Ca' Foscari University, Venice, Italy, **2** Department of Environmental Sciences, Informatics and Statistics, Ca' Foscari University, Venice, Italy

☯ These authors contributed equally to this work.
* andrea.gasparetto@unive.it

Abstract

Text Classification methods have been improving at an unparalleled speed in the last decade thanks to the success brought about by deep learning. Historically, state-of-the-art approaches have been developed for and benchmarked against English datasets, while other languages have had to catch up and deal with inevitable linguistic challenges. This paper offers a survey with practical and linguistic connotations, showcasing the complications and challenges tied to the application of modern Text Classification algorithms to languages other than English. We engage this subject from the perspective of the Italian language, and we discuss in detail issues related to the scarcity of task-specific datasets, as well as the issues posed by the computational expensiveness of modern approaches. We substantiate this by providing an extensively researched list of available datasets in Italian, comparing it with a similarly sought list for French, which we use for comparison. In order to simulate a real-world practical scenario, we apply a number of representative methods to custom-tailored multilabel classification datasets in Italian, French, and English. We conclude by discussing results, future challenges, and research directions from a linguistically inclusive perspective.

Introduction

Text Classification (TC) is one of the most essential tasks in the field of Natural Language Processing (NLP). This denomination is usually associated with a broad category of more specific procedures, which roughly share the common objective of designating predefined labels for a given input body of text. Over the years, TC procedures have evolved from simple, rule-based systems to highly specialized architectures. The latter have gone closer than ever before to showing actual understanding of the underlying semantics of a piece of text, utilizing such meaning in order to make an informed decision for the classification process.

There are countless practical applications of TC, including information retrieval, topic labeling, sentiment analysis, and news classification. Even more loosely related tasks, such as extractive text summarization and content-based recommendation systems, can be approached within a TC framework.

OPEN ACCESS

Citation: Gasparetto A, Zangari A, Marcuzzo M, Albarelli A (2022) A survey on text classification: Practical perspectives on the Italian language. PLoS ONE 17(7): e0270904. https://doi.org/10.1371/journal.pone.0270904

Editor: Sathishkumar V E, Hanyang University, KOREA, REPUBLIC OF

Received: January 14, 2022

Accepted: June 18, 2022

Published: July 6, 2022

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pone.0270904

Data Availability Statement: All data and code used for the experimental session have been linked in the article, except for Reuters Corpus Volume 1 and 2, which, as a dataset owned by a third party, cannot be shared, but must be requested to the

National Institute of Standards and Technology (NIST). The Reuters Corpus Volume 1/2 is a collection of Reuters News stories that can be used for research purposes. Volume 1 contains only English articles, while Volume 2 contains articles in 13 different languages. Articles are annotated with various hierarchical labels, including topic codes which have been used as a general descriptor of the article's content. All information on how to request this data are described on the following website: https://trec.nist.gov/data/reuters/reuters.html. Once data has been obtained, the procedure to generate the dataset and reproduce the presented experiments is described in the code repository available here: https://gitlab.com/distration/dsi-nlp-publib#rcv1-rcv2-news-categorization.

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Due to the speed at which textual information is produced, it has become essential to rely on automatic processing techniques to handle continuously increasing volumes of data. However, the adoption of modern machine learning (ML) methods in this context can be non-trivial. Recent ML methods rely on the ingestion of massive amounts of textual documents in order to effectively model a probability distribution over sequences of words. Hence, the limited availability of text corpora (i.e., large collections of digitized textual data) in some languages constitutes a serious obstacle to the application of these methods. Such resources are essential to the development of modern approaches and add to the intrinsic difficulties of this task.

Resource categorization of language

In order to better understand how the resources tied to a language influence the application of ML algorithms, we briefly discuss the topic of resource categorization of languages. In the field of computational linguistics, it is common to define language as low-, mid- or high-resource. While there is no standardized approach to determine whether a language fits into one category or the other, a reasonable categorization is usually easy to find and agree on. The resources described by these denominations refer to raw data (i.e., collections of digitized text) as well as linguistic tools and software necessary to perform various tasks. In the particular context of Text Classification, tools like these might be needed to perform common text interpretation procedures such as lemmatization and part-of-speech (PoS) tagging (further outlined in the Preprocessing section).

As there is no standardized approach to this classification, the spectrum of resources in which languages lie is highly speculative, but it is fair to claim that the most well-resourced end is dominated by English and Chinese (Mandarin). Other languages commonly considered as high resource include Arabic, French, German, Portuguese, Spanish and Finnish, though most of research implicitly utilizes one of the former two languages, English in particular [1].

Throughout this survey, we utilize Italian as a means of comparison. As a language, Italian can be considered on the higher end of this spectrum, somewhere between a mid- and high-resource categorization. Indeed, as far as raw computerized text data is concerned, it is a rather well documented language. However, task-specific data, indispensable to test and validate TC algorithms, can be severely lacking for this language. Moreover, sets of textual data for specific downstream tasks may only be available if duly licensed (and sometimes at a cost), something that is certainly true for Italian as well as many other languages. Obviously, this can vastly limit the potential for research.

The importance of generalizing linguistic research

Much of modern research focuses on a few, dominant high-resource languages like English. However, something that is certainly desirable for any NLP model is for it to be validated on its capability of generalizing its result on data and languages other than those on which it has been trained and tested [2]. This concept is relatable to that of *language independence* [3], an attribute that describes models that can be made to work comparably well across languages.

Because this aspect is often overlooked, applying Text Classification procedures can be challenging in languages other than English. This might be because linguistic tools are lacking (or simply perform differently), or because suitable benchmarks are not readily available. Furthermore, as research has moved more and more toward deep learning methods, the divide can be further exacerbated by the interpretability issues of these models, which makes it difficult to ascertain their effectiveness on other languages.

Major differences and contributions

In recent years, multiple excellent works have reviewed TC from a generic, language-agnostic perspective. Li et al. [4] provide a comprehensive investigation of models ranging from traditional approaches to more deep learning-based models. We follow their excellent categorization of approaches. Kowsari et al.'s [5] survey is notable for its in-depth exploration of stages such as feature extraction and dimensionality reduction, which are more common in traditional approaches. Minaee et al.'s [6] work focuses solely on a thorough exploration of deep approaches, though it notably also provides quantitative results for classical methods in its experimental performance analysis. The main objective of this work is to provide insight into the main linguistic challenges involved in the development of TC methods as applied to languages other than English. While we provide a brief summary of some of the most prominent TC techniques, we emphasize those aspects related to the linguistic component of this task.

The main language studied while surveying these methods was Italian; to reiterate, this is a well-documented language, for which we will however showcase a scarceness of task-specific datasets. To this end, we provide a list of notable TC datasets for Italian and complement it with a similarly built list for the French language, such as to provide a fair comparison with a high-resource language that is not necessarily English. We describe how to distill a multilabel dataset for topic labeling from Wikipedia dumps, as well as a news classification dataset from Reuters articles. We perform a study of compatibility between a set of representative algorithms and these two datasets for the multilabel classification task, for which we discuss various challenges and difficulties encountered. In summary, this study's main contributions are as follows:

- We provide a high-level overview of TC, highlighting which steps of the pipeline have been shown to be more language-dependent;
- We highlight recent developments in classification methods for NLP, including modern preprocessing operations and pre-trained language models;
- While introducing the operation of a TC pipeline, we discuss the main causes of compatibility issues with languages other than English;
- We demonstrate the applicability of several traditional and modern methods to multilabel datasets in three different languages (Italian, French, and English);
- We underline technical challenges and the current research directions being explored to solve them.

The rest of this survey is organized as follows. The first section discusses text preprocessing, going into detail about language segmentation as the most relevant operation from a linguistic point of view. We then discuss text representation techniques utilized to project preprocessed text into a feature space, briefly describing early methods and how they evolved into contextualized and semantically meaningful vectorial embeddings. We discuss the issues posed by the computational expensiveness of these methods, and why these are problematic for their application in multiple languages. We dedicate a short section to classification algorithms and how their importance has diminished in favor of better text representation. The latter sections deal with experimental factors, describing TC tasks and showcasing datasets in Italian and French, outlining the search criteria, and providing a selection of English datasets for comparison purposes. We provide quantitative results for a select choice of multilabel datasets in all three languages. Finally, we summarize the main future challenges faced by TC methods, before concluding the survey. Datasets and code used for the experimental part of this work are available (when legally possible) at https://gitlab.com/distration/dsi-nlp-publib.

Preprocessing

A fundamental part of the Text Classification pipeline resides in its preprocessing steps. Raw textual information is *unstructured* and does not have a straightforward numerical representation (differently, for example, from types of data such as images). Clearly, from a linguistic point of view, languages are indeed ruled by a very complex structure, one that might be intuitive to a native speaker of that language, but much less so to a machine.

It becomes therefore necessary to project text into an appropriate feature space so that it can be handled by a learning algorithm. In this section, we discuss all those procedures that prepare textual data for this projection, whether this is done through manual feature extraction (as with earlier, more traditional methods) or automatically (as is with recent, deep learning-based approaches). We provide an overview of the most important preprocessing operations, while the section that follows will describe possible choices for obtaining machine-friendly representations from the resulting preprocessed text. We place particular focus on tokenization as, among the early steps of language interpretation, it is certainly the most critical, having a considerable impact on downstream performance on several NLP tasks [7].

Tokenization

The first and most basic operation is that of *tokenization*, the process of breaking a stream of text into smaller chunks (historically called *tokens*). The most traditional as well as intuitive atomic unit of choice (i.e., token) has been centered around words [8]. Recent approaches have instead been applying more granular decomposition processes, such as character *n*-grams, sub-words and, most recently, even segmentation approaches based on the underlying byte representation of text [9]. It has been argued that, among preprocessing operations for any NLP task, tokenization can be regarded as the most important language-dependent operation [10].

The following sections will describe the main difference between more traditional and recent tokenizers, showcasing an interesting trend towards maximal decomposition. It is worth mentioning that, as of now, researchers agree that there is no single best solution, and the choice of unit of text is one that must be made depending on the context and necessity of the application.

Pre-tokenizers. Conventional approaches to the tokenization task have traditionally been *rule-based*, and, especially in most white-spaced writing systems (i.e., languages where spaces are used as word separators in writing), minimal tokenization can be carried out by separating around blank characters, punctuation, and contractions. Clearly, this intuitive approach has seen many refinements, often integrating language-specific knowledge into its rules. While not perfect, such segmentation approaches are deemed an acceptable approximation of actual morphemes, striking a compromise between linguistic irrelevance and purely typographic tokens [8]. Examples of popular rule-based tokenizers include Moses [11], and the SpaCy tokenizer [12]. Both Moses and SpaCy are NLP toolkits and include tokenizers that work with multiple languages using a set of language-specific rules and exceptions.

Recent literature often defines earlier tokenization approaches as “pre-tokenizers”, because of how many modern methods may use them as an initial step (therefore preceding “proper” tokenization).

Data-driven tokenizers. Tokenization (and language segmentation in general) has evolved greatly in recent decades. Here, we introduce some of the latest developments in the field, such as to highlight their close relationship with language representation approaches.

When provided with textual data, a tokenizer will decompose it and create a "vocabulary" of terms. At a practical level, this vocabulary is used to generate an index-based mapping between actual tokens and a numerical representation (different depending on the feature extraction technique). Modern text representation techniques are based on *embeddings*, rich vectorial representations which we will cover in the Text representation section. As each token in the vocabulary corresponds to a possibly large embedding, these representations are unable to handle arbitrarily vocabularies of arbitrary size because of time and space limitations. As a consequence, most modern language representation techniques require a fixed-size vocabulary.

It is clear, then, that modern tokenization approaches must strike a balance between the expressiveness of the vocabulary and its dimension. This expressiveness is most closely tied to the concept of out-of-vocabulary (OOV) words, which correspond to text units that have not been seen during a model's training. As such, the model is unable to extract useful information from OOV tokens (models such as these are termed as *closed-vocabulary*) [8]. A sufficiently expressive vocabulary, then, should be able to minimize the number of OOV terms, such as to fully utilize the information available at inference time.

OOV words are a central weakness of traditional tokenization approaches. Because of phenomena such as derivations, inflections, and contractions, certain languages can be difficult to segment properly, creating excessively large vocabularies. A solution can be to reduce tokenization to a character-level segmentation; while this has been tested with some degree of success, in many languages it can be hard to obtain a meaningful representation for single characters since they appear in too many different contexts and are not as relevant as, for example, words in terms of sequence modeling [13]. Furthermore, since each character is mapped to its own vector of parameters, the memory footprint increases for longer sequences. Many modern neural language representation approaches resort to truncation of input sequences to a pre-defined length in order to handle memory issues; doing this with character tokenization would mean keeping the first *k* characters instead of the first *k* words, potentially losing much of the original sequence information.

As both of these simple strategies are not entirely satisfactory, modern tokenization approaches most commonly employ hybrid techniques that split text into *sub-words*. Notably, while manually constructed approaches to this type of segmentation have been tested, the more popular method of choice for recent methods relies on automatically learning morphological segmentation in an unsupervised manner. The general idea of data-driven tokenizers is that frequently used words should not be split into smaller words, while rare words should be broken into more "reusable" fragments; this way, OOV tokens can be recognized as a composition of multiple known sub-words. In the following paragraphs, we introduce some of the most popular tokenizers that have seen widespread adoption in modern NLP models in recent years. Table 1 provides a concise view of the main modern tokenizers.

Byte Pair Encoding. An important breakthrough in tokenization strategies was the development of Byte Pair Encoding (BPE) [14], originally proposed as a data compression algorithm [19] and later adapted for sub-word segmentation. After a character-level pre-tokenization, smarter tokenization is learned by iteratively computing the co-occurrence of consecutive pairs of vocabulary terms, and merging the most frequent into a new vocabulary word. The same process is then applied when tokenizing unseen documents, executing recorded merges in the same order as they were during training. A notable extension of this segmentation

Table 1. Most widely adopted recent tokenization approaches.

Tokenizer	Training Procedure	Inference Procedure	Language Support
BPE [14]	Merge most frequent consecutive pairs of *n*-grams	Merge incrementally, keeping merged term if in vocabulary	White-spaced only
BBPE [15]	Same as BPE, based on bytes instead of *n*-grams	Same as BPE	All languages
WordPiece [16]	Merge sub-words that maximize LM likelihood	Find longest first substring of words within vocabulary	White-spaced only
UnigramLM [17]	Start from pre-generated vocabulary, remove sub-words that least contribute to the LM likelihood function	Substring likelihood maximization through Viterbi Algorithm	All languages
SentencePiece (sw package) [18]	Fast, optimized procedures for other algorithms	Enhanced inference methods	All languages

https://doi.org/10.1371/journal.pone.0270904.t001

procedure is byte-level BPE [15], which applies the same algorithm not to characters but to raw bytes.

WordPiece. The WordPiece tokenizer [16] was initially developed for Japanese text segmentation problems, and relies on the creation of *n*-gram-based language models (in the classical sense, as we describe later in the Text representation section) to recognize recurring syllables, prefixes and word segments in a corpus. A greedy process iteratively increases the vocabulary size, starting from single characters, selecting and merging pairs of sub-words that maximize the language model likelihood. The algorithm stops when the expected likelihood falls below a predefined threshold, or the maximum vocabulary size is reached.

UnigramLM. Conceptually similar to WordPiece, UnigramLM [17] proceeds in the opposite direction, starting from a large vocabulary obtained by pre-tokenization and iteratively removing the terms with the lowest expected probability with regards to a simple unigram language model. The process is repeated until the desired size is reached. Multiple segmentations are possible due to the stochastic nature of this process, and while the most likely segmentation is chosen in practice, it is possible to implement sampling procedures to perform what is defined as "sub-word regularization", which has empirically been shown to improve results on some tasks.

SentencePiece. SentencePiece [18] is not an algorithm in itself but rather a software package containing optimized versions of the above approaches. Among other segmentation optimizations, it is a particularly worthy mention as it addresses the fact that other tokenizers depend on knowing which characters act as word separators in the corpus, which is language-dependent and may require specific pre-tokenization procedures to create rules to recognize word boundaries. Instead, SentencePiece considers text as a raw stream of characters, including word-separators, removing this operational constraint.

Linguistic aspect of tokenization. The segmentation of textual data into sentences and words has been historically rooted in linguistic motivations (as well as technical constraints). The common and intuitive approach of segmenting into words has the advantage that, from a linguistic point of view, these units can be labeled with linguistic annotations such as PoS tags (e.g., noun, verb) and syntactic dependency information (related to the structure of sentences) [8]. Therefore, utilizing linguistically motivated units opens the possibility of using such additional information throughout the classification pipeline.

However, it is not trivial to define and identify linguistic units, most notably because of the vast number of irregularities and language-specific phenomena involved. Works such as the Morpho-Syntactic Annotation Framework (MAF) ISO standard [20, 21] identify linguistic units as *word-forms*: these are represented by a stem and a list of inflections to be attached. For example, many English words can be inflected as verbs, adverbs, nouns, and adjectives. Word-forms cover many linguistic phenomena, such as contractions (e.g., "*isn't*"), compounds (e.g., "*football*"), morphological derivatives (e.g., "*sadness*"), diminutive or augmentative derivations and more. Nevertheless, deriving a precise procedure to segment into word-forms is hard and expensive, and word-based segmentation is usually accepted as a reasonable approximation.

Similarly, other works focus on morpheme-based tokenization for morphologically complex languages [22–24]. Morphemes are the indivisible basic units of language that carry semantic meaning; learning meaningful context-independent representations of morphemes is challenging, particularly for agglutinative languages, where words can be composed by (almost) arbitrarily long and complex sequences of morphemes with minimal contextual change. This is in contrast to fusional languages, where morphemes are stitched together usually with more radical adaptations [25]. For example, the Turkish agglutination "*evlerden*" can be seen as the composition of a stem and two word elements, "ev-ler-den", meaning "from the houses", composed by a concatenation of morphemes translating literally to "*house*-(*plural modifier*)-*from*". Clearly, simple white-space tokenization will not suffice in the recognition of these three morphemes.

Modern sub-word tokenization strategies, as discussed, put the linguistic significance of tokens aside. Tokens in the vocabulary are instead selected using model-based approaches that require an appropriate amount of training data but do not rely on explicit language-specific knowledge. In other words, these tokens are not seeking to have a one-to-one correspondence to morphemes, and may also span through different words, depending on the co-occurrence of character sequences in the training corpus.

Sub-word segmentation. As mentioned, traditional tokenization procedures often approximate linguistic units as an acceptable compromise. Modern segmentation procedures, on the other hand, often do not have explicit linguistic motivations or explanations and are instead based on automatic learning processes, trained for efficient tokenization on large unlabeled corpora. Unsupervised word segmentation with neural models has seen particular interest in languages that are notoriously difficult to segment because of their lack of white-space delimiters (Chinese, Japanese) or because of their highly productive morphologies tokens (Arabic, Hebrew) [26]. Reducing the number of OOV terms is particularly important for the latter case, as downstream tasks such as classification would incur too high a loss of information if they were just removed. However, it has been argued that languages such as these, as well as agglutinative languages, may be better served with character-level models or small sub-word inventories [27, 28], even though sub-word segmentation has reasonable motivation [8]. Cases like these reinforce the notion that there is no single best solution for language segmentation.

Maximal decomposition. As previously mentioned, some recent proposals have proposed maximal decomposition of text based on its underlying bytes rather than typographical tokens (e.g., words, sub-words, characters). An example widely used in recent models is that of byte-level BPE, which applies the BPE compression algorithm on bytes rather than characters [15]. This is not only a compact representation (up to 256 possible values for a vocabulary), but crucially agnostic to languages, and has seen success in languages particularly difficult to segment. Encoding byte-level representations is not however as simple as it may seem, as byte sequence representations are often much longer than character sequences. Moreover, as Mielke et al. [8] point out, byte-level modeling is not necessarily unbiased; while characters are intrinsically tied to language representation, different character encodings are unrelated to linguistics. For example, Unicode-based representations were not created with linguistic motivations, and different languages may have different representations (for example, might require multiple bytes per character). Another approach being explored is that of "visual" modeling, utilizing the pixels that compose the graphical representation of text, which may be promising for languages with rich visual features (e.g., Chinese, Korean) [29].

Shared vocabularies. Many NLP applications must be able to handle text in different languages simultaneously. It is possible to utilize a number of language-specific tokenizers, but shared vocabularies have also been proposed for multilingual systems. These systems work with a vocabulary composed of a variable number of word segments derived from different languages. Thus, there is no language-specific set of recognized tokens, but only an expanded multilingual vocabulary. As can be expected, a same token might be shared across different languages: in this case, its vectorial representations will have to encompass its meaning in multiple languages. While the sharing of learned representations is enticing, inconclusive results have been found in this regard [30]. Moreover, recent language representation approaches based on shared-vocabulary tokenization tend to be biased towards high resource languages such as English (even when oversampling low-resource languages), propagating this bias to downstream tasks such as TC [31].

A related work by Rust et al. [7] evaluated the performance of several monolingual tokenizers pre-trained on monolingual corpora and reports results in terms of two custom-tailored metrics. In their research, which concerns the effect of tokenization strategies on downstream tasks when paired with recent approaches, they explore the difference in performance between monolingual and multilingual tokenizers. The former are based on prior research on monolingual models and are mostly based on the WordPiece algorithm. However, many of these tokenization strategies rely on additional language-specific preprocessing. Examples include Japanese utilizing a pre-built morphological parser whose tokens are then split into characters, Arabic testing pre-segmentation techniques before applying the WordPiece algorithm, and Korean introducing bi-directional conditioning in the WordPiece algorithm. The authors found that the multilingual tokenizer performed inconsistently across languages, producing a lower number of tokens in morphologically poor languages and over-segmenting the richer ones. The latter are more challenging because root morphemes are frequently enriched with affixes to match the context of the sentence, including grammatical gender, case, number, or person. This translates to a higher number of possible combinations of words that require either more data or language-specific tokenizers. The issues and challenges faced by multilingual tokenizers and the shared vocabularies they produce are an active area of research [8].

Summary. Text segmentation is a fundamental part of any NLP task, with high linguistic relevance and important ramifications throughout the pipeline of a classifier due to its intrinsic ties to the embedding creation process. The previously mentioned work by Rust et al. [7] reports that the tokenization strategy (and, relatedly, the size of training data) are among the greatest driving forces for downstream task performance. They also found that utilizing monolingual tokenizers in multilingual models can lead to improved performance in most tasks and languages.

There is much more to be said about this topic, and we point interested readers to the work by Mielke et al. [8] which provides a comprehensive dissertation on the issues of tokenization strategies and emphasizes the limits of fixed vocabulary data-driven tokenizers, including the ones related to bias in data and language fairness for multilingual models.

Other preprocessing operations

In this section, we briefly outline other common preprocessing operations applied to already tokenized text. Notably, most modern tokenizers already apply a number of the noise removal and “soft” normalization processes we will describe (e.g., lowercasing). Other more “destructive” operations, which remove or alter words altogether, should instead be considered carefully and on a case-to-case basis, as modern NLP models are typically trained to extract context from grammatically and morphologically sound sentences and performance will likely suffer if they are applied to heavily preprocessed corpora with a very different distribution of words.

Noise removal. The set of tokens produced by tokenization might contain unnecessary or misleading elements, such as superfluous symbols or characters. *Noise removal* refers to the set of operations used to remove those tokens and words that are deemed unnecessary or harmful to solve a specific task. Such procedures may also include lowercasing, misspelling correction, and standardization of slang words and abbreviations, which are all intended to reduce the number of different elements to be projected in the feature space. Earlier approaches commonly resorted to the removal of stopwords, non-informative words with no discriminative importance for classification and that are common in languages (e.g., articles, pronouns, etc.) [32, 33].

Stemming and lemmatization. As traditional text interpretation approaches are unable to capture significant semantic information about words, a further simplification of the feature space can (and has been shown to) be beneficial [34, 35]. Therefore, simplifying words by reducing inflections to a common form can be helpful in relating words that earlier methods would otherwise be unable to tie together (e.g., “child” vs “children”). This is most commonly achieved through either *stemming* or *lemmatization*, which derive the stem or lemma (canonical form) of a word, respectively.

Linguistic considerations. Many relevant linguistic aspects were already covered when discussing tokenizers and language segmentation in general, which is easily the most influential preprocessing step. Other operations, such as stemming and lemmatization, also have a similar linguistic connotation, but in a more traditional sense; indeed, they are generally rule-based or vocabulary-based, meaning that they are specifically created to process text in a language and depend on a manually defined set of rules and common affixes, stopwords and base lemmas. For example, the SpaCy rule-based lemmatizer uses a set of cascading rules that reduce tokens to a base form, according to possible word-forms that are applicable to the recognized PoS. The language-specific vocabulary is then used to determine if the lemmatized word actually exists.

Each language requires specific adaptations of rules and vocabularies to perform noise removal operations, which can notably have varying success in different languages. The reasons behind these differences in performance between languages are likely to be attributable to differences in the complexity of morphology that are more difficult to model through a set of rules. Additionally, vocabularies for rule-based procedures might potentially be incomplete because of the high variance in the number of lemmas in different languages.

For instance, lemmatizing a document in Greek is likely to be much harder than lemmatizing a document in Italian; as an empirical example, the SpaCy documentation reports a much lower score for the former’s lemmatization accuracy [36, 37].

Text representation

Following a preprocessing procedure, a body of text will be transformed into a list of separated, standardized tokens that might have been through multiple filters. Before it can be understood by a computer, however, it must be expressed in numeric form. Feature representation techniques are a fundamental part of any NLP application, many times trumping in importance the actual classification step of the overall pipeline. In this section, we give a brief overview of the most frequently utilized traditionally, and segue to a discussion of recent approaches.

Language modeling. An important concept in text representation is that of *language models* (LMs). These are a statistical representation of text which has been studied for decades, though it has seen a new rise in popularity due to the application of deep neural models.

Intuitively, language models aim to predict the likelihood of a string given a preceding or surrounding context (usually a sequence of words, or, more in general, tokens). The related task is referred to as *language modeling*.

Formally, a statistical language model can be described as a probability distribution over sequences of words. Given a sequence of words $s = w_1, w_2, \ldots, w_m$, the model assigns a probability $P(w_1\, w_2 \ldots w_m)$ to the whole sequence. While the goal is to assign probabilities to whole sequences of words, the task is related to that of computing the probability of an upcoming word and is framed as such. *N*-gram models are a simple example, making use of the Markov assumption, by which the prediction probability is based only on the last *n* words before it—i.e., $P(w_1 w_2 \ \ldots \ w_n) = P(w_i | w_{i-n} \ \ldots \ w_{i-1})$. Traditional algorithms such as Maximum Likelihood Estimation (MLE) [38, 39] can be used to solve the probability prediction task. The probability scores given are context-specific (in general, they relate to a better-structured sentence, such as a good translation).

From text to vectors

We introduce in this section the most influential strategies for the representation of text sequences. We start by outlining the more traditional approaches for the conversion of text to numerical form, which are based on word occurrence frequency, and move on to the more advanced methods which utilize the underlying idea of language modeling.

Bag-of-Words. Traditionally, the most basic representation of text has been that of *Bag-of-Words* (BoW) [40–42]. As the name suggests, this model reduces bodies of text to unordered collections of words in which sentence structures and semantic relationships between its elements are ignored (hence, the intuitive visualization as a "bag of words"). Though simple, this approach has been widely used throughout ML applications (even outside of NLP, where it is commonly referred to as "Bag-of-Features") [41, 43–45]. Furthermore, it is common to utilize a feature extraction technique such as Term Frequency (TF), which maintains the relative frequency of words in a single vector for the entire text rather than a one-hot encoding of each word. This is usually paired with an Inverse Document Frequency (IDF) [46] factor, which penalizes common words within the entire corpus of texts (since they do not help discriminate between them). Vocabularies generated by TF-IDF representations may encounter time and memory complexity issues. One possible solution is to limit the maximum number of features represented (in practice, pruning low-scored words) or, alternatively, a dimensionality reduction algorithm can be applied. Popular approaches which have seen success include Principal Components Analysis (PCA) [47], Linear Discriminant Analysis (LDA) [48] and Non-Negative Matrix Factorization (NMF) [49].

Word embeddings. Earlier methods focused on capturing the syntactic representations of words but lacked the capability of encapsulating semantic meaning inferred from context. For example, they possessed no way to assimilate word synonyms. In the last decade, researchers have proposed to leverage language modeling to produce *word embeddings* as a solution to this problem. Intuitively, this self-supervised feature learning technique is aimed at learning a mapping between each piece of text (most commonly words, hence the name) to a *n*-dimensional vector of real numbers. These approaches are based on shallow neural networks, which learn these mappings through different learning procedures; in general, they are based on the assumption that the meaning of a word can be extracted from its surrounding words in a sentence. Some of the most popular and effective word embedding techniques based on these principles are Word2Vec [50], GloVe [51] and FastText [52].

Differently from BoW representations, which are only concerned with word occurrence statistics, this latter technique can embed much more information in the learned representation, depending on the objective of the training procedure used to generate it. In the simplest case, word embedding techniques produce representations based on the surrounding tokens: similar contexts produce similar embeddings. This is generally the idea behind pre-trained embeddings, that are released for general usage and are not designed for specific tasks.

However, it should be noted that many strategies can be devised to enrich learned embeddings with more discriminative features, sometimes by fine-tuning pre-trained ones on other tasks. For instance, Qin et al. [53] propose a different approach, which uses two neural networks to learn features from randomly initialized embeddings. Features extracted by the first module are projected to the orthogonal direction of their counterpart, in order to learn more relevant, uncommon features. This can be seen as similar to the idea used to generate contextualized embeddings, which we describe in the next section.

Deep language models. Word embeddings represent an important milestone toward the creation of neural language models. As said, shallow learning-based architectures such as Word2Vec focused on the embeddings rather than the model itself, creating largely "static" (i.e., context-independent) vectorial representations. The denomination of embeddings as static can be attributed to the fact that polysemous words (words with more than one meaning) map to a same embedding, which can therefore be understood as a combination of the multiple senses of such word the model has encountered during the training process. From a practical point of view, these embeddings work like lookup tables, where every recognized word is mapped to a single fixed-size vector that sums up all of the contexts that a particular word has appeared in during training. Hence, for a given word to be encoded, the output vector is the same, no matter its context in the sentence.

A variety of deeper architectures have been applied to TC using pre-trained word embeddings, in an effort to improve the models' capacity and to create more meaningful semantic representation. Among other enhancements, Autoencoder frameworks and Recurrent Neural Networks have been particularly influential. The latter are prime candidates in the modeling of sequential data, and they have been applied with success to word embedding techniques [54, 55].

Contextualized embeddings. As more complex and deep architectures were applied to improve the learning process of text representation, some researchers proposed to use deep NN-based LMs to add context to word embeddings. This contextualization process is an augmentation of the previously described "static" embeddings. To obtain a *contextualized embedding* for a word, the static one is passed through a model that transforms variable-length sequences of left- and/or right-context words into a single fixed-length vector. Hence, unlike previous static word vectors, these embeddings are generated from both the static ones and the parameters of a contextualizer LM, producing distinct embeddings even for the same words used in different contexts [56]. The relation between contextualized embeddings and static embeddings is shown in Fig 1. In particular, note how the embedded sentence enters the model in its entirety, allowing the model to contextualize individual representations based on the surrounding tokens. Studies have shown that layers in deep language models are specialized to capture different linguistic information [57].

While this approach was first tried with recurrent-based LMs, notably ELMo and ULMFiT [58, 59], it has been rendered ubiquitous by the introduction of purely attention-based models, made popular by the seminal Transformer architecture [60] and the subsequent development of the Bidirectional Encoder Representations from Transformers (BERT) [61] and the Generative Pre-trained Transformer (GPT) [62]. Among other advantages, such deep models are able to benefit from increasingly large numbers of parameters, usually achieved by multiplying the number of layers in their architecture, something which was crucially not the case for recurrence-based models [63]. These LMs, which we informally term *contextualized LMs*, are hence

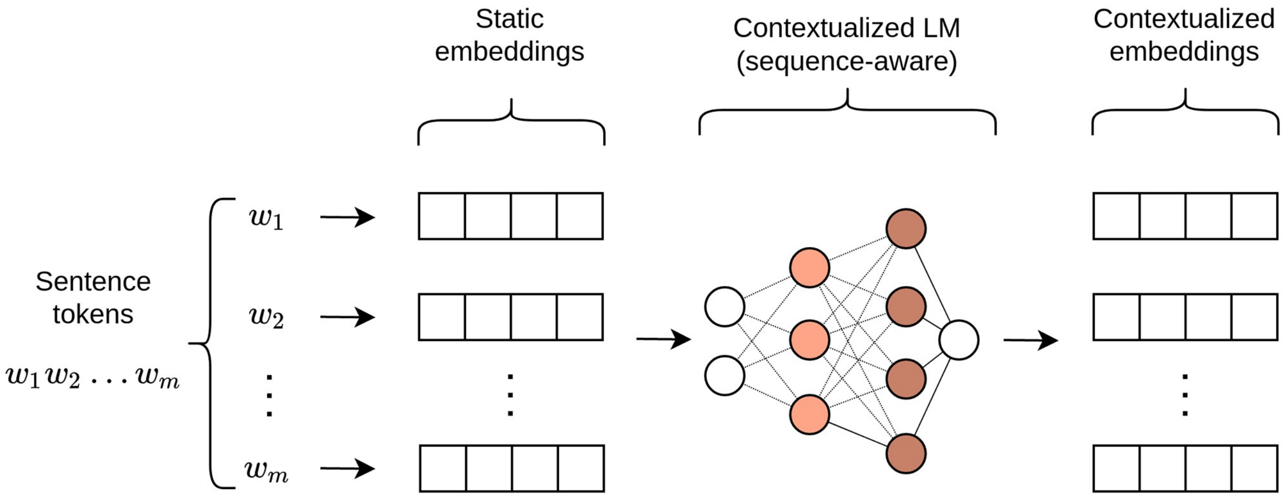

Fig 1. Sample generation process for contextualized embeddings.

https://doi.org/10.1371/journal.pone.0270904.g001

able to disambiguate polysemous words by looking at the surrounding tokens in the sentence. Conversely, one should note that contextualized embeddings are not meant to be extracted "statically", i.e., as with a one-to-one mapping from word to vector. Instead, the language model should always be provided the surrounding context of a word in order to produce a meaningful word vector.

Contextualized LMs are usually pre-trained on a language modeling task (e.g., next word prediction) and are used as transfer-learning methods in other NLP tasks [64]. Adaptation to tasks is typically carried out through fine-tuning of the model, or part of it, on domain-specific data. Various strategies have been proposed, depending on the base model, one of them being training the backbone model with a task-specific head on top of it, possibly even freezing the backbone parameters.

Feature extraction in other languages

In this section, we contextualize text representation techniques to their utilization in languages other than English, highlighting the challenges of training LMs and closing the section with an overview of models trained on Italian corpora.

Traditional approaches in other languages. As earlier approaches were not capable of expressing the semantic meaning of words, they can be largely seen as "detached" from language (with most linguistic aspects falling on tokenization approaches, as discussed). Therefore, the performance of methods such as BoW or TF-IDF relies largely on preprocessing and principled usage of statistical methods. Since not meant to really understand languages, their utilization does not see much or any difference when used on different ones (though their performance might vary depending on their specifics). In contrast, word embeddings are pre-trained on large, usually monolingual corpora, and are thus specific to the language represented within the data.

Popular examples of such corpora are Wikipedia dumps [65] and the Common Crawl archive [66], whose size allows for more robust and generic representations. It is possible to manually train these embeddings from scratch (provided that the dataset is of sufficient size) or fine-tune a set of pre-trained ones; both approaches aim to enrich vectors with dataset-specific knowledge. Embeddings specialized on the domain data could (and usually do) result in better performances in downstream tasks such as TC.

It is hard, however, to determine how much data is required to meet the "sufficient size" criteria, since it specifically depends on the task at hand and the quality of the data. This adds to the fact that learning word embeddings is a long and computationally expensive procedure. Because of this, it is common to utilize pre-trained, open-sourced embeddings as a starting point. In the context of mid-resourced languages such as Italian, pre-trained word embeddings can usually be obtained reliably. A lower-resourced language might have to resort to manual training of these embeddings, which may require non-trivial computational resources—a topic we will address in more detail in the following sections.

Contextualized language models in other languages. Transformer-based language models have revolutionized how NLP solutions are sought. Unlike its predecessors, this generic methodological approach is applicable to a wide variety of tasks, often needing very little work to specialize it towards the specific downstream problem. As mentioned, however, the majority of research is done in the English language, which is taken as a "good representative" for the applicability of its results. Adaptations to other languages are created at different speeds and degrees, as the development of contextualized language models is made difficult by their high computational complexity and necessity for large amounts of data.

In the following, we will illustrate how impactful these requirements can be for practical development, analyzing the challenges and possible solutions being developed, as well as going into detail about the resource landscape for the Italian language.

Pre-training of contextualized language models. Pre-trained language models based on deep learning need to be trained on large corpora of text in order to achieve a good generalization capability. For instance, the original implementation of BERT was trained on BooksCorpus (800M words) [67] and English Wikipedia (2,500M words). The authors emphasize the necessity of utilizing document-level corpora, such as to extract long contiguous sequences which lead to better generalization. Finding these types of resources is much easier in the case of English, but mid-resourced languages such as Italian usually have access to sufficient resources for pre-training on self-supervised tasks. Indeed, this particular challenge will affect more severely low-resourced languages, rather than mid-resourced ones.

Recent research has begun to take different directions when it comes to pre-training approaches, attempting to either specialize or generalize pre-training data. While not in the context of classification but rather that of summarization, Zhang et al. [68] showcase a higher performance when data and learning objectives utilized in pre-training more closely mirror the final task of the overall system. This is in contrast to the generic approach of other language models, which are in many ways agnostic to downstream applications in favor of generality. Conversely, the recent GPT-3 [69] model tries to leverage massive datasets and processing power to create a model generic enough to overcome the need for specialized approaches. In particular, the GPT-3 is meant to address the issue of small downstream datasets, showing promising results for approaches with low label rates.

Both of these developments reveal insights into what we can expect future challenges to be. In the first case, researchers have empirically shown better results with specialized data; as we will showcase in the section discussing our findings on task-specific datasets, that kind of data is not as easy to come by as general-purpose, task-agnostic text information. In the second case, the difficulties are tied to the vast computational expenses, which we will discuss in the next section.

Notably, research has also pushed towards methods that are able to generalize well between languages. XLM-R [70] is a RoBERTa-based model pre-trained on more than 2 terabytes of unlabeled corpora in more than 100 languages. The result is a multilingual pre-trained model suitable for fine-tuning on a variety of multilingual and monolingual tasks. The authors reported competitive performance with respect to monolingual models. The development of multilingual and multipurpose language models suggests the possibility of future research possibly converging towards fewer, more inclusive contextualized language models. Nevertheless, all language models gain much from the massive size of the datasets they are trained on, and the availability of such corpora is still problematic for under-represented languages. Moreover, we have highlighted how language-specific tools such as monolingual tokenizers may still be beneficial to downstream task performance, questioning their one-to-one replacement with purely multilingual approaches.

Computational resources. The computational resources required to develop contextualized language models of the BERT and GPT families are, without a doubt, incredibly high. Many recent evolutions of these models that have been proposed have tens of times the number of parameters of the original ones. While not a linguistic challenge per-se, it is evident that the conspicuous computational requirements will also act as entry barriers, preventing widespread research in this area.

While many authors do not disclose the actual training times and hardware infrastructure utilized by these models, it is safe to estimate that the upward trend in processing power required will continue [71]. The larger the models, the higher the number of trainable parameters for the network, translating into often prohibitive costs of development. Computational complexity is clearly challenging because of multiple aspects, spanning from environmental concerns (tied to the amount of power consumed to produce models that are substituted every year) to the fact that it is becoming more and more prohibitive to perform proper experimentation because of the cost of a single training procedure.

To put it into perspective, it is sufficient to consider the aforementioned GPT-3 model, whose largest iteration flaunts close to 175 billion parameters, which amounts to more than 1500 times the trainable parameters of BERT's base model. Researchers have estimated a positive correlation between number of parameters and model performance [72], theoretically justifying the push for larger and larger models. More recently, GPT-3 has been surpassed in size by even bigger models, such as Google's GShard [73] and Switch-C [74], which have 600 billion and 1.6 trillion parameters respectively. Fig 2 shows a visual representation of this trend in recent models.

While the examples provided are not representative of all Transformer-based language models, the lack of computational resources can be problematic even in more common scenarios. An example can be made of the fine-tuning procedure, which needs to be performed for any downstream task; however, working with such large models—even if pre-trained—might still be challenging just because of how expensive it is to load them into memory. Again, this severely limits the possibility for experimentation and evaluation in different languages.

Reducing the cost of Transformer-based LMs. In response to this issue, researchers have devised much more compact models which are still able to achieve similar results, while being considerably more applicable in practice. DistilBERT [75] leverages knowledge distillation to reduce the size of BERT models by up to 40% while retaining 97% of its effectiveness. TinyBERT [76] extends knowledge distillation to the task-specific learning stage. A different approach is proposed by models like ALBERT [77], which introduces parameter-reduction techniques to reduce the memory consumption and increase the training speed of BERT models. Similarly, ELECTRA [78] introduces a more sample-efficient pre-training task in place of masked language modeling (MLM), namely token detection.

In practical scenarios, whenever processing power is limited, utilizing downsized models such as these might be a solution. This is especially true for models such as ALBERT and ELECTRA, as they devise clever ways to improve the efficiency of the pre-training task, while DistilBERT and TinyBERT still require the original model as a "teacher" in the distillation process. It is also possible to perform a fine-tuning procedure without involving the language

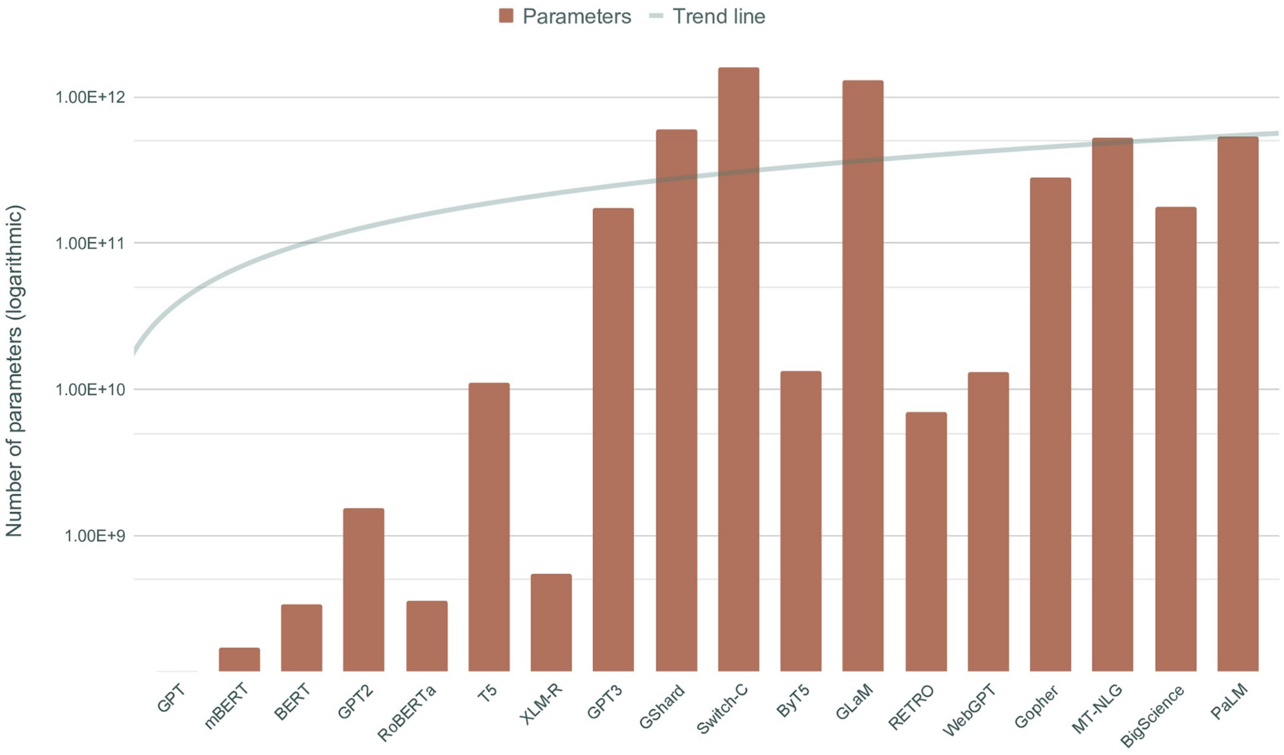

Fig 2. N. of parameters (log scale) in the largest version of recent contextualized LMs, ordered by release date.

https://doi.org/10.1371/journal.pone.0270904.g002

model in the learning process (i.e., "freezing" the base model's weights). This is more akin to utilizing the underlying word embeddings in their agnostic state (but still contextualized). The computational resources necessary are therefore vastly reduced, though this severely shrinks the learning capacity of the overall system.

Language models in Italian. With the revolution brought about by BERT-derived models and their successors, researchers have quickly begun to study their applicability in specific tasks. In this section, we highlight a few of the studies made for Italian, as well as some multilingual approaches.

Tamburini et al. [79] studied the performance of BERT-like models in classic NLP tasks, such as PoS-tagging, NER, and universal dependency parsing, as well as some considerations on sentiment analysis. They highlight the most prominent pre-trained models available at the time of writing and find them to allow for a large increase in performance for almost all of them. An example of these works is ALBERTo [80], an Italian model based on a slightly modified BERT architecture and trained on tweets for sentiment analysis tasks. GPT models have also been adapted, with works such as GePpeTto [81], a GPT-2 based model for Italian—though it is evaluated on generative tasks rather than classification.

As was briefly mentioned, multilingual approaches have also been studied, mostly on Multilingual BERT [61], a LM trained on the concatenation of monolingual Wikipedia corpora from 104 languages. Pires et al. [82] devise a zero-shot cross-lingual model transfer, in which the model is fine-tuned for a downstream task in one language and tested for that same task in a different language. The results of their experiments demonstrate that the model is able to generalize to different languages (including Italian) quite well, though it performs best on typologically similar languages. Nevertheless, the best performance is still achieved by fine-tuning on the target language, hence suggesting that it is preferable when possible. While multilingual adaptation is possible, authors argue that deeper fine-tuning is needed when compared to monolingual approaches, especially whenever the task is more related to semantics [82, 83].

Existing models in Italian. Through platforms like Hugging Face [84] and Tensorflow [85], pre-trained language models based on various architectures are made available for multiple languages. As it was for previous word embeddings, it has become common practice to open source such models because of how long and expensive their training procedure is. Whenever the computational resources are not available, it becomes necessary to rely on the contributions of others, which may not be as plentiful in all languages. Table 2 showcases some of the pre-trained models that are available for the Italian language at the time of writing. Minor changes like case sensitivity or vocabulary size differences are excluded, while the annotation "M" stands for "multilingual". Whenever the "# of parameters" column has multiple entries, it refers to the various model sizes available (usually, a smaller "base" model and a "large" one).

Theoretically, better results are to be expected if the domain of the downstream task (e.g., news articles) is contained in the pre-training dataset; however, this is an aspect that is becoming less and less important, as large amounts of data usually yield better results regardless. Pota et al. [83], who analyze the performance of various models on a Twitter sentiment analysis task, report that a generic BERT model pre-trained on large, general-purpose corpora of plain text can outperform a model pre-trained entirely on tweets like ALBERTo, even though the latter is trained on a corpus that is closer to the one used in the final task. The authors attribute this result to the size difference between pre-training datasets.

Summary. The state-of-the-art approach to the projection of text into a feature space involves the creation of contextualized language models, which must be utilized at inference time to extract context-specific embeddings on which to perform, for example, classification tasks. While not specific to classification, many studies have shown how this is the most important step of the pipeline, and therefore one that requires considerable attention. Recent trends have gone towards more and more costly models, which, in response, are largely open-sourced to allow practitioners with fewer resources to make use of these approaches. However, as shown, rigorous development and experimentation still require those resources in order to be performed, vastly limiting the possibilities for those who do not have such computing capabilities.

Table 2. Italian pre-trained transformer models.

Name	Paper	Source	Architecture	# of parameters
Italian BERT	-	[86]	BERT	110M
AlBERTo	[80]	[87]	BERT	110M
Italian ELECTRA	-	[86]	ELECTRA	110M
UmBERTo	-	[88]	RoBERTa	110M
GilBERTo	-	[89]	RoBERTa	110M
GePpeTto	[81]	[90]	GPT-2	117M
Recycled GPT-2	[91]	[92]	GPT-2	117M / 345M
Multilingual BERT (M)	[61]	[93]	BERT	172M
XLM-Roberta (M)	[70]	[94]	RoBERTa	270M / 550M

(M) Multilingual model.

https://doi.org/10.1371/journal.pone.0270904.t002

Classification step

So far, we have described preprocessing and feature representation approaches, fundamental to Text Classification but at the same time shared with a wide variety of NLP tasks. The importance of appropriate text representation cannot be understated; in fact, as previously stated, recent approaches have shown outstanding results with very simple classifiers, further cementing the notion that effective projection of text into an appropriate feature space is essential.

In this comparatively shorter section, we highlight the changes in how classification is tackled in traditional and recent approaches. Unsurprisingly, many end-to-end classifiers, especially neural ones, are largely based on effective feature representation, further supporting the idea that semantic understanding of the language is at the base of any NLP task. An informal —yet intuitive—explanation of this result is that understanding the content of a body of text is the most important step in the classification pipeline, much like a person would likely be able to label a piece of text if it understood what it meant.

Traditional classification methods

Traditional learning models put a large focus on preliminary data preparation and feature engineering phases. While this is also true for modern models, earlier approaches required much more aggressive preprocessing, with a much higher dependency on the removal of noise and unimportant words that added no discriminative power to the pipeline. This can be challenging, as languages encompass a large and varied amount of rules of dependencies. Nonetheless, after a set of features has been extracted, it is possible to apply generic classification approaches. As they are generic, it is hard to attribute any real language-specific insight to them.

For the sake of completeness, we provide in Table 3 a high-level view of a number of traditional TC approaches. For a more in-depth description of these methods, we point to Kowsari et al.'s [5] survey. It is worth mentioning that these methods, still have a place in practical uses for TC—certainly in environments with small or very specific datasets, where injections of domain-specific knowledge in preprocessing steps and feature handcrafting may be relevant.

Table 3. Traditional classification techniques.

Model	Advantages	Disadvantages
Rocchio Classifier [95]	Simple and computationally cheap	Lacks robustness, not well suited for multiclass classification or multimodal classes
Naïve Bayes [96]	Easy to implement and train, fast calculation process	Strong feature independence assumptions
Conditional Random Fields [97]	Flexible feature design, combining advantages of classification and graphical modeling	High computational complexity and issues with online learning
Hidden Markov Models [98]	Well-studied approach, suitable for sequentially ordered bodies of text	Strong assumptions typical of probabilistic methods
k-Nearest Neighbors [99]	Non-parametric, fast under the right conditions, easy adaptation to multiclass	Unfavorable scaling with high-dimensional spaces, choice of *k* is arbitrary, a distance function between text bodies is hard to define
Support Vector Machines [100, 101]	Effective non-linear modeling even in high dimensional spaces, robust against overfitting	High memory complexity and requires a non-trivial decision of a kernel function, not transparent, does not produce probabilities directly
Decision Trees [102]	Naturally models categorical features, fast and interpretable	Very susceptible to noise and overfitting, weak against diagonal decision boundaries
Logistic Regression [103]	Easy to implement and train, does not necessitate re-scaling of features or fine-tuning	Strong independence assumption of data points, only suitable for linear problems
Random Forests [104]	Fast ensembling approach, reduces variance of single decision trees	Loss of interpretability and inference speed, still prone to overfitting
Ensembles [105, 106]	Collection of classifiers are more robust and accurate, less prone to overfitting	Expensive training, difficult interpretation and careful fine-tuning is required

https://doi.org/10.1371/journal.pone.0270904.t003

Neural methods

The necessity for classical models to handcraft features has, over time, proven to be especially limiting. Due to the strong dependence between these features and the domain itself, good feature engineering often necessitates extensive domain knowledge. In turn, this makes approaches difficult to generalize to new tasks and languages.

The development of word embeddings therefore marks an important paradigm shift. Much of the work done by deep approaches is in fact towards automatic extraction of semantically meaningful representations from text. In this section, we provide an outline of how neural models, based largely on deep learning, have evolved in recent years, highlighting a trend where much of the focus is on text representation. This section does not aim at giving a comprehensive overview of the discussed neural architectures, as they are not the focus of this work. We refer to the surveys by Li et al. [4] and Minaee et al. [6] for a more comprehensive coverage.

Multilayer Perceptrons. In the earlier years of adoption of deep learning models, researchers developed deep neural networks based on simpler architectures, such as Multilayer Perceptrons (MLP), which displayed good results thanks to their ability to capture latent features automatically [107, 108]. Such models usually treat input text as an unordered Bag-of-Words, where input words are represented through some feature extraction technique (like TF-IDF or word embeddings). However, some of these approaches attempt to integrate further information about the syntactic structure of text, with examples such as Paragraph-Vec [108], which incorporates the syntactic ordering of words as well as the contextual information of paragraphs.

Recurrent Neural Networks. More influential, however, were architectures based on Recurrent Neural Networks (RNNs), as the ability to interpret text as sequences of tokens allows them to capture latent relationships between contextual words [109, 110].

In general, a simple RNN for text processing is fed a sequence of word embeddings, that are processed one at a time. At each time step, the model receives the next word vector and the hidden state of the previous time step. Standard RNN architectures are most frequently enhanced with more advanced gating mechanisms, the most popular being the Long Short-Term Memory (LSTM) [111] and Gated Recurrent Units (GRU) [112]. These enhancements address many of the gradient-related issues faced by vanilla RNN frameworks. The introduction of bidirectionality in RNNs has also been proven beneficial [113] and has been applied to LSTMs, with notable results such as ELMo [58], a language modeling approach that relies on BiLSTMs and is one of the first milestones in the development of contextualized word embeddings.

Among the most utilized approaches, encoder-decoders based on recurrence [54] have been particularly influential. The hidden layers of these architectures implicitly learn a semantically and syntactically meaningful representation of text that can be used for classification. On the downside, recurrent models have inherent limitations due to their sequential nature, as sequentiality precludes parallelization. Longer sentences can also run into memory constraints and, more crucially, are seen as RNNs true bottleneck because of how the network tends to forget earlier parts of the sequence, making for an incomplete representation [114].

Convolutional Neural Networks. Convolutional Neural Networks (CNNs), though most commonly used in the field of computer vision, have also seen applications in the context of NLP and TC [115]. The most straightforward application has convolutional filters applied over word embeddings, most commonly with size as wide as the embedding dimensionality, as to always consider the entire vector representation for each word. The main upside associated with CNNs is their speed and how efficient their latent representations are. Conversely, other properties that could be exploited while working with images, such as location invariance and local compositionality [116, 117], make little sense when analyzing text. Many approaches have been proposed, one of the most popular being TextCNN [118], a comparatively simple CNN-based model with a one layer convolution structure that is placed on top of word embeddings.

Graph Neural Networks. In the last few years, graph representations have seen a resurgence in various fields of AI [119, 120]. In particular, Graph Neural Networks (GNNs) have received increasingly more attention [121], and this has also been the case for TC. GNNs utilize graph structures to capture dependencies and relations between their nodes.

Numerous well-established approaches to neural networks have been generalized to arbitrarily structured graphs. Among them, the convolution operation—usually applied to regular grid structures [122, 123]—is particularly popular because of its effectiveness and convenience [6]. Convolutions propagate information between nodes, and consecutive convolutions allow the network to spread the information further away, providing an effective way to model higher-order connectivity. Recently, successful approaches have been obtained on heterogeneous graphs in which nodes are both words and documents; TC is thus cast as a node classification task for document nodes.

The real strength of graph networks comes from their feature extraction capabilities, with examples such as TextGCN [124]. Both word and document embeddings are learned through convolutions. Researchers have also tried to train BERT and GCN models jointly, as in [125].

GNNs are among the few architectures able to compete with contextualized language models in downstream tasks and can perform quite well in low label rate datasets [124]. Some of the major weaknesses of graph-based approaches reside in model complexity, which becomes an issue with large-scale text corpora due to memory limitations. Simplified models such as SGC [126] help in this regard, while also mitigating one of the other major issues of GNNS, that of *oversmoothing*—where node representations converge to a same value and become indistinguishable [127].

Transformer-based language models. The already mentioned Transformer, proposed by Vaswani et al. [60], is considered the most recent major breakthrough in sequence processing methods and especially in NLP. The Transformer architecture is based on an encoder-decoder framework with multiple attentive blocks stacked together. Crucially, the main novelty resided in the removal of any recurrence-based layers for modeling sequentiality, instead relying on the *attention mechanism* alone. For further details, we point to surveys such as Gasparetto et al. [128] and Li et al. [4]. Transformer-based methods have built on the original architecture while maintaining the same basic principles, and have significantly boosted the performance achievable on various NLP tasks. During pre-training, these models are able to encode generic linguistic knowledge that can be transferred to any downstream task via a fine-tuning procedure on task-specific data.

Some of the most influential contextualized language models are based on research that suggests that limiting the architecture to either encoders or decoders may result in equivalent performance and lighter models [129]. According to this principle, the previously mentioned Generative Pre-trained Transformer (GPT) [62] utilizes an architecture based on stacking multiple transformer-decoder layers, resulting in an autoregressive model that is trained on a unidirectional next word prediction task. While particularly suitable for generative tasks, it has also been successfully adapted to TC. On the other hand, the seminal Bidirectional Encoder Representations from Transformers (BERT) [61] relies instead on a multi-layer bidirectional Transformer encoder architecture. This model employs specifically tailored learning tasks—in particular, masked language modeling (MLM) and next sentence prediction (NSP)—in order to incorporate bidirectional conditioning. The adaptation of BERT to downstream tasks is very simple. In fact, outstanding results have been achieved in classification by simply fine-tuning a model that passes the representations obtained by the encoders through a single-layer, feed-forward neural network. It is common to allow this training procedure to also affect the representation learned by the pre-trained language models, such as to specialize the overall model on the domain of the task being faced. In practice, the changes to the language model parameters (i.e., everything that precedes the classification head) are minimal; this is desirable since if it were otherwise the language model would incur too great a loss of generality.

BERT and GPT laid the foundation for many variants and improvements to their original framework. Among them, we cite the Robustly optimized BERT approach (RoBERTa) [130], which explores the importance of hyperparameter choice and improves its learning procedure, and the GPT-2 [131], which instead improves mostly in terms of data utilized and scale of the models. More recent developments have also been proposed, both in terms of architecture and in scale (i.e., size of pre-training data and number of model parameters). See [128] for a more exhaustive coverage of the latest LMs. When discussing future research directions and challenges, we will highlight some of the most relevant to the topics of this survey.

Summary of language factors in the TC pipeline

As mentioned at the start of this section, much of the focus of the classification pipeline has shifted towards effective text representation. Contextualized language models are able to perform exceedingly well across different tasks (including TC) with very simple classifiers—most frequently, a simple feed-forward layer. Crucially, researchers have reported that these results can be obtained even without large amounts of fine-tuning and parameter optimization, such as in the work by Tamburini [79], which studies these phenomena in the Italian language. We also found this to be true in our experiments with two Italian labeled corpora, as will be outlined in later sections.

We wrap up the overview of classification methods by drawing some conclusions on the overall classification pipeline, as viewed when considering different languages. We highlighted the importance of tokenizers; proper text segmentation is fundamental to the feature projection step and therefore has direct impact on the final downstream performance of tasks such as TC. Language-specific tokenization strategies have been shown to have advantages over generic, language-agnostic approaches. Still, many multilingual language models have relied on data-driven tokenizers, like BPE and WordPiece, achieving remarkably good results without being rooted in linguistic knowledge. An excellent example is that of ByT5, a recent multitask transformer model which follows the byte segmentation approach previously mentioned and obtains state-of-the-art results [132]. Nonetheless, despite a few efforts in this direction, the impact of language-specific preprocessing on large language models has not been thoroughly explored.

We have also showcased how language modeling with contextualized representation has obtained outstanding results, further cementing the idea that effective semantic representations of text are arguably the most important phase of any NLP task. However, we mentioned how computation complexity can be particularly daunting. Utilizing pre-trained language models is certainly advantageous and effectively democratizes their adaptation to many downstream tasks, but may entail a certain "rigidity" in their adoption. For instance, it should be reminded that models such as BERT or GPT are closed-vocabulary; replacing the tokenization strategy is not possible without performing again the entire pre-training procedure. Therefore, testing how the performance of a downstream task is affected by changes within the classification pipeline is likely to be an expensive process.

Finally, we have highlighted how some researchers are experimenting whether models pre-trained with task-specific objectives and data can outperform general-purpose models. As of now, results seem inconclusive, and there is no clear indication of whether this approach (which is clearly much more complex) will be preferred to the generic approach.

Datasets

The availability of annotated corpora is essential for NLP research. While the development of deep language models mainly leverages self-supervised strategies, labeled datasets are required for supervised tasks like TC. In this section, we provide a comprehensive list of resources available in two European languages and compare them with the resources available in the English NLP research area. We decided to focus our search on annotated corpora in Italian, which we regard as a mid-resource language, and French, which we instead consider a high-resource language.

While a consistent number of written English annotated corpora is available and heavily referenced in the literature, we find that the quantity of easily accessible resources in the languages we considered is still lacking (especially in Italian), limiting research on this theme. To reiterate, this is a fundamentally different issue from the one posed by low-resource languages, usually characterized by insufficient unlabelled data to even be able to perform self-supervised procedures, but it is by no means a less important one.

Overview

Text classification tasks. We conduct a scientific literature search of annotated textual corpora presented or employed in research contributions. Reflecting Li et Al. [4], we focus on the following common TC sub-domains:

- *Sentiment analysis (SA)*: the task of understanding the emotional content of a piece of text, usually mapping it to predefined categories representing specific emotions. We include in this category the tasks of stance and polarity detection, as well as the identification of rhetorical devices, like irony, or linguistic properties, like subjectivity;
- *Topic labeling (TL)*: the task of extracting the topic (or theme) of a document, for example an article. This task is often related to content recommendation, since it can be used to map textual contents to user interests;
- *News classification (NC)*: classification of news into specific categories, like user interests or topics;
- *Question answering (QA)*: extractive question answering can be framed as a classification problem where the model, given a list of candidate sentences extracted from text and a target question, must decide which sentence contains the answer;
- *Natural language inference (NLI)*: given a pair of statements, the task is to determine if one is entailed by the other;
- *Named entity recognition (NER)*: the task of locating and classifying named entities mentioned in unstructured text into predefined categories;
- *Syntactic parsing (SP)*: the task of predicting the morpho-syntactic properties of words in sentences, like part-of-speech (PoS) tagging, speech dependencies, and semantic role labeling.

These tasks can be adapted to different domains, and many sub-tasks with different formulations are possible. They are commonly used as benchmarks in NLP research, especially as part of multitask natural language evaluation initiatives, like the General Language Understanding Evaluation (GLUE) benchmark [133].

Search criteria. In order to balance search time and effectiveness, our search strategy is composed of three steps:

1. Search for datasets on Google Dataset Search (https://datasetsearch.research.google.com);
2. Search for publications on Google Scholar using keywords along the lines of "Italian text corpus" and "Italian Text Classification".
    a. The first two pages of results sorted for pertinence are explored;
    b. The same is repeated by filtering results based on their publication date, by looking at contributions published from 2019 onwards;
3. Search on PapersWithCode (https://paperswithcode.com/datasets) for contributions using the same keywords.

For every publication, we explore all referenced publicly accessible data sources.

Datasets in other languages

In this section we present the results of our search, omitting corpora that are not public or that are unavailable at the time of our search. We further filter out datasets with less than a few thousand labeled samples, unless they are highly specialized datasets that we deem potentially valuable for ML applications.

Italian and French datasets. We list monolingual corpora for Italian and French in Tables 4 and 5. Table 6 describes multilingual classification corpora containing one or both of these languages, and possibly others. We mark with a single asterisk (*) datasets available through a request for access. Additionally, we mark with (**) datasets that are not distributed for free or require specific affiliation. When defining tasks, we use the abbreviations introduced in the previous section and otherwise use TC to indicate a generic "Text Classification" task that does not fit any of the defined categories. For the sake of comparison, we give an estimate of the dataset size, based on the published documentation. Size can be expressed as the number of labeled sentences ("S"), tokens or words ("T"), or documents ("D") available in the corpus, and is comprehensive of all training, test, and evaluation splits. For multilingual datasets in Table 6, it refers to the number of samples in Italian and French (or, when similarly sized, an average of the two).

English datasets. A comprehensive list of English TC datasets is provided by Li et Al. [4]. In order to make a comparison, we report in Table 7 some of the most popular English datasets, along with their size and related tasks. In this specific case, our search is limited to popular datasets used within PapersWithCode recent contributions.

Findings

Many of the Italian datasets listed in Table 4 were created for the EVALITA initiative, a periodical campaign organized by AILC for the evaluation of NLP and speech tools for the Italian language. The most recurrent tasks proposed in this initiative fall into the sentiment analysis and syntactic parsing domains. While most datasets assembled by participants in this initiative are made openly available and provide great value to the Italian NLP research, they are often small in comparison to English datasets for the same task, and always fewer in number. SemEval is a similar international workshop for the evaluation of semantic analysis systems [240].

Table 4. Italian datasets.

Name	Paper	Source	Task	Size	Unit
ABSITA	[134]	[135]	SA	10,000	S
SENTIPOLC	[136]	[137]	SA (irony, subjectivity)	9,400	D
ATE_ABSITA	[138]	[139]	SA, TL	4,300	D
AMI 2020	-	[140]	SA (misogyny)	9,900	S
R-ITA	[141]	[142]	SA (stance)	1,000	D
ChroniclItaly	[143]	[144]	NER, SA	8,600	D
IHSC	[145]	[146]	SA, SP	6,900	D
HaSpeeDe	[147]	[148]*	SA, SP	8,500/3,500	D
SQuAD-it	[149]	[150]	QA	61,000	D
Fact-Ita Bank	[151]	[152]*	NER, SP, NC	65,000	T
FLaIT	[153]	[154]*	NER, SP	1,500	S
PAISÀ	[155]	[156]	SP	250,000,000	T
KIPoS	[157]	[158]*	SP	200,000	T
iLISTEN 2018	[159]	[160]	SP	22,000	T
PoSTWITA	[136]	[161]	SP	6,700	D
TUT	[162]	[163]	SP	3,500	S
TE-EVALITA 2009	[164]	[165]	NLI	800	D
GxG	[166]	[167]	TC (gender)	11,000	D
DaDoEval	[168]	[169]	TC (date)	2,800	D
AcCompl-It	[170]	[171]*	TC (acceptability, complexity)	1,680/2,530	S
ITAmoji	[172]	[173]*	TC (emoji prediction)	275,000	D

* Available through a request for access.

https://doi.org/10.1371/journal.pone.0270904.t004

Table 5. French datasets.

Name	Paper	Source	Task	Size	Unit
French Twitter SA	-	[174]	SA	1,500,000	D
Allociné	-	[175]	SA	200,000	D
French Sexism Detection	[176]	[177]	SA (sexism)	11,800	D
Event2018	[178, 179]	[180]*	SA (stance), TC (event)	15,000/137,000	S
CAS	[181]	[182]*	SP, SA (uncertainty, negation)	4,900	D
E-FRA	[141]	[142]	SA (stance)	2,000	D
FQuAD	[183]	[184]*	QA	26,000	D
PIAF	[185]	[186]	QA	3,800	D
Quaero Broadcast News XT	-	[187]**	NER	1,500,000	T
Quaero Old Press XT	-	[188]**	NER	1,300,000	T
La Recherche	-	[189]**	SP	447,000	T
FTB	[190]	[191]*	SP	644,000	T
ParisParl	-	[192]	SP, TC (affiliation)	203,000,000	T
French Corpus MWE	[193]	[194]	SP	166,000	T
French FraCaS	[195]	[196]	NLI	346	D

* Available through a request for access.

** Require payment or specific affiliation.

https://doi.org/10.1371/journal.pone.0270904.t005

Table 6. Multilingual datasets.

Name	Paper	Source	Languages	Task	Size	Unit
Webis-CLS-10	[197]	[198]	Fr, En, +2	SA	69,000	D
Amazon Reviews ML	[199]	[200]	Fr, En, +4	SA	210,000	D
SemEval-2016 Task 5	[201]	[202]	Fr, En, +6	SA, TL (aspect)	2,400	S
Reuters Corpus Volume 2 (RCV2)	[203]	[204]*	It, Fr, +11	NC	28,406/85,393	D
MLSUM	[205]	[206]	Fr +4	NC	425,000	D
KB Europeana Newspapers NER	-	[207]	Fr +3	NER	-	-
WikiAnn	[208]	[209]	It, Fr, +280	NER	7,5 mln	T
DAWT	[210]	[211]	It, Fr, En, +3	NER, EDL	1,5 mln	D
WikiNER	[212]	[213]	It, Fr, En, +6	NER, SP	260,000	S
NewsReader MEANTIME	[214]	[215]	It, En, +2	NER, SP, NC	15,000	T
Universal dependencies	[216]	[217]	It, Fr, En, +100	SP	∼1 mln	T
XL-WiC	[218]	[219]	It, Fr, +1	SP	2,000/70,000	S
Aranea	[220]	[221]	It, Fr, +20	SP	120 mln/1,2 bln	T
PANACEA	[222]	[223]	It, Fr, +2	SP	-	-
MKQA	[224]	[225]	It, Fr, En, +23	QA	10,000	D
CLEF QA Test Suites	-	[226]**	It, Fr, En, +7	QA	160,000	D
XLNI	[227]	[228]	Fr, En, +13	NLI	7,500	D

* Available through a request for access.

** Require payment or specific affiliation.

https://doi.org/10.1371/journal.pone.0270904.t006

Multilingual datasets provided for the proposed tasks tend to be small and, while access is provided, they are not easy to find and use outside the context of these initiatives.

We hereby discuss the availability of task-specific datasets as compared to analogous English counterparts (Table 7). One should note that some of the datasets presented, especially in French, are made available through the ELRA-ELDA catalog (available at http://www.elra.

Table 7. English datasets.

Name	Languages	Task	Size	Unit	Reference
20 Newsgroup	En	NC	18,800	D	[4, 229]
Reuters	En	NC	10,700	D	[4, 230]
R8	En	NC	7,600	D	[4]
R52	En	NC	9,100	D	[4]
RCV1	En	NC	804,000	D	[203]
AG News	En	NC	127,600	D	[231]
TREC-6	En	QA	5,500	D	[232]
SQuAD 2.0	En	QA	150,000	D	[233]
Yahoo!Answers	En	QA	1,460,000	D	[234]
Yelp-2 / Yelp-5	En	SA	8,600,000	D	[235]
Amazon-5	En	SA	3,650,000	D	[4]
Amazon-2	En	SA	4,000,000	D	[4]
IMDb	En	SA	50,000	D	[236]
DBpedia	En +124	TL	630,000	D	[4, 237]
MultiNLI	En	NLI	433,000	D	[238]
CoNLL-2003	En +1	NER	301,000	T	[239]

https://doi.org/10.1371/journal.pone.0270904.t007

info/en). This resource requires an ELRA membership plan and/or the payment of a fee in order to be accessed.

Syntactic parsing. Corpora for syntactic parsing (SP), like PoS-tagging and lemmatization, are well resourced in both languages reviewed in this paper and are featured in several multilingual treebanks (like Universal Dependencies [216] and Panacea [222]). Furthermore, the PAISÀ corpus stands out as a large monolingual dataset in Italian for these tasks.

News classification. On the other hand, we noticed a lack of news classification (NC) datasets in Italian and French. The only notable exceptions are the MLSUM and RCV2 datasets. The MLSUM multilingual dataset contains news extracts labeled with their summaries and topic, and it is available in French but not Italian. On the other hand, the Reuters RCV2 dataset for multilingual news classification contains both Italian and French sub-corpora. This dataset can only be accessed by sending a request and signing an organizational agreement.

Topic labeling. Likewise, topic labeling (TL) datasets are also scarce in Italian and French. Wikipedia dumps and DBpedia represent a remarkable source of crowdsourced semi-structured data that can be employed for topic labeling and TC in general and are available in hundreds of languages. However, labels must first be extracted and merged following some criteria which have not yet been standardized. Though these corpora have already seen use in the literature [241, 242], there is no consistent set of annotations to be used as reference. While categories assigned to Wikipedia pages by contributors are often used as predictive targets, these frequently contain spurious or improper information that can be treated in many different ways, or should arguably be removed entirely.

Sentiment analysis. Multiple sentiment analysis (SA) datasets are available in Italian and French—often targeting user-generated content—for detection of polarity, political stance, or specific rhetorical devices (like irony). More than ⅓ of the Italian datasets are scraped from social or e-commerce platforms, especially Twitter.

Question answering. There is at least one large question answering (QA) dataset for both Italian and French, as well as several multilingual ones which contain both languages. For this task, the size of these datasets is comparable to the main English QA datasets.

Named entity recognition. Similarly, there are multiple large multilingual corpora for named entity recognition (NER) tasks, at least for the most classic formulation of this task aimed at recognizing "person", "location" and "organization" entities.

Semantic entailment. We found two semantic entailment (NLI) datasets containing the Italian and French languages, the largest containing 1,000 and 7,500 samples respectively. By comparison, one of the most popular English NLI datasets (MultiNLI) is more than 50 times the size of the French one mentioned.

Cross-lingual benchmarks

Multitask evaluation benchmarks like GLUE, SUPERGLUE (for English), and FLUE (for French) are increasingly popular tools to evaluate models across a wide variety of NLP tasks. This incentivizes models to share knowledge across different tasks and gain sufficient language understanding to generalize on a wide range of applications. Notably, the recent publication of the XTREME and XGLUE benchmarks introduced support to multilingual and cross-lingual cross-task evaluation. Still, for some tasks, not all languages are available. For example, among the classification tasks we are interested in, XGLUE supports only PoS-tagging (included in our SP category) and web page ranking on the Italian language. Additionally, they do not provide training data in every language, and, for some tasks, data is extracted from other multilingual datasets (namely XNLI, Universal Dependencies, and WikiAnn). Table 8 summarizes popular language-specific and cross-lingual benchmarks.

Table 8. Linguistic benchmarks.

Benchmark	Paper	Languages	Tasks	Cross-Lingual
GLUE	[133]	En	QA, SA, TL, NC	
SUPERGLUE	[244]	En	TC, NLI, QA	
FLUE	[245]	Fr	TC, SP, NLI, PAR*	
XTREME	[246]	It, Fr, En, +37	NER, SP, NC, QA	✓
XGLUE	[243]	It, Fr, En, +8	NER, SP, NC, QA	✓

* Paraphrasing.

https://doi.org/10.1371/journal.pone.0270904.t008

The goal of these initiatives is not to provide more monolingual data for languages with fewer resources, but rather to encourage the evaluation of cross-lingual models capable of transferring knowledge across different languages, even those with little or no training data. Their contribution is important in that it provides a standardized evaluation environment for deep learning models that could alleviate the common low resource issue [243].

Applicability evaluation

We have previously mentioned the rising computational costs of developing state-of-the-art NLP solutions. In this section, we simulate a practical case by synthesizing two custom multilabel classification datasets. In our research, we have found that multilabel TC in the Italian language (and, to some degree, in French) are understudied, in contrast to binary and multiclass TC. In a similar vein to Tamburini [79], our study is aimed at gauging how easily and how well these methods can apply to new tasks and datasets with a constrained amount of resources (i.e., only fine-tuning).

Datasets utilized

We use one monolingual dataset per language for each studied task. We decided to tackle the multilabel classification task, as it was more interesting for our research work and, to some extent, is less documented in the literature. An empirical evaluation of the labels in our TL dataset finds that the categorizations utilized are overlapping and cannot be easily decomposed into binary sub-classification tasks (e.g., "sports" vs "winter sports") [247]. A similar consideration can be made of the NC dataset, which was multilabel by construction.

We chose to synthesize these datasets mainly because of the scarcity of other options in Italian. In the first case, we found that synthesizing a TL dataset from Wikipedia was the only way of obtaining a reasonably large, general-domain, annotated corpus in Italian. Similarly, the RCV2 dataset was chosen for the NC task because no other public collection of annotated news articles was available in Italian.

Topic labeling. We synthesized a dataset for the Topic Labeling task using Wikipedia dumps in all three languages. For each dump, articles and related topics are extracted using a modified version of the popular WikiExtractor tool (see https://github.com/attardi/wikiextractor). After an exploration of the data, we came to the conclusion that Wikipedia categories are ill-suited for a topic labeling task since they are often too specific and hardly provide a good topic indication. Therefore, we decided to use a different approach, and annotate extracted articles with the Wikipedia portals they are assigned to.

Currently, there are roughly 500 portals within the English version of Wikipedia, while there are more than 500,000 categories. Wikipedia itself states that portals serve as entry points for articles that belong to the same broad subject [248], thus making them better targets for our task. Our final datasets contain only the 100 most populous portals, and article frequency has been limited to a maximum of 50,000 articles per label. This was done both to contain the dataset size and to reduce class imbalance.

News classification. For the News Classification (NC) task we utilized the Italian and French subset of articles in the Reuters multilingual collection (RCV2), as well as the English monolingual collection contained in RCV1. The English Reuters collection has been for a long time one of the staple TC corpora utilized for experimental purposes [247, 249], though its multilingual version has not seen as much attention. The articles are not "parallel", in the sense that they do not contain the same content in different languages, but are different articles altogether.

The RCV1/2 articles are labeled with a variety of tags that describe their content at varying degrees of specificity. The most consistent and interesting tags across languages were topic codes; within such codes, subjects are ordered in a hierarchical manner. However, articles are often tagged inconsistently—the depth of hierarchy within tags ranges between two and four levels, and documents are sometimes only partially tagged. We decided to retain topics at the second level of specificity; each article is tagged with all second-level topic codes it contains and stripped of any other. Only topics that had at least 500 representatives were included in the final datasets. Articles are deprived of all topics excluded this way, and the article is discarded if it contains no topics after this process.

Analysis of datasets. Final statistics of the described datasets are reported in Table 9. The main difference between the TL and NC datasets is the length of the articles; an average over the number of tokens per document reveals Reuters articles are comparatively much shorter than extracted Wikipedia articles. Indeed, Wikipedia articles are usually much more descriptive, while Reuters articles are presented in a very concise and to-the-point format. As the LMs we worked with allow for a maximum of 512 tokens as input, we can expect a truncation to be much more frequent in the case of Wikipedia articles. The information loss should however not be dramatic, as most discriminative information is usually found at the beginning, where the article is introduced.

Figs 3 and 4 depict the distribution of the number of topics per article for the ItWiki-100 and RCV2it datasets, respectively. The same statistics for the French and English counterparts are supplied in the S1 Datasets supplement. Unsurprisingly, most articles have few labels (between one and three), with a large amount having only one label. The larger Wiki datasets have a longer tail-shaped distribution, with a few outliers having a large number of labels, but that overall make up a small part of the datasets (for instance, articles with four or more labels make up less than 1.4% of the ItWiki-100 dataset).

We further report in the S1 Datasets supplement the distribution of topics, i.e., the number of articles labeled for each specific topic. All datasets follow a similar distribution, with a number of well-represented classes and a lower bracket of classes that are comparatively underrepresented. We point out that class imbalance was much more severe in the raw data we

Table 9. Statistics on the examined datasets.

Name	Classes	Avg n. tokens	Samples	Task
ItWiki-100	100	354	892,573	TL
FrWiki-100	100	362	1,494,761	TL
EnWiki-100	100	741	329,626	TL
RCV2it	15	123	25,750	NC
RCV2fr	38	224	79,173	NC
RCV1en	57	216	758,149	NC

https://doi.org/10.1371/journal.pone.0270904.t009

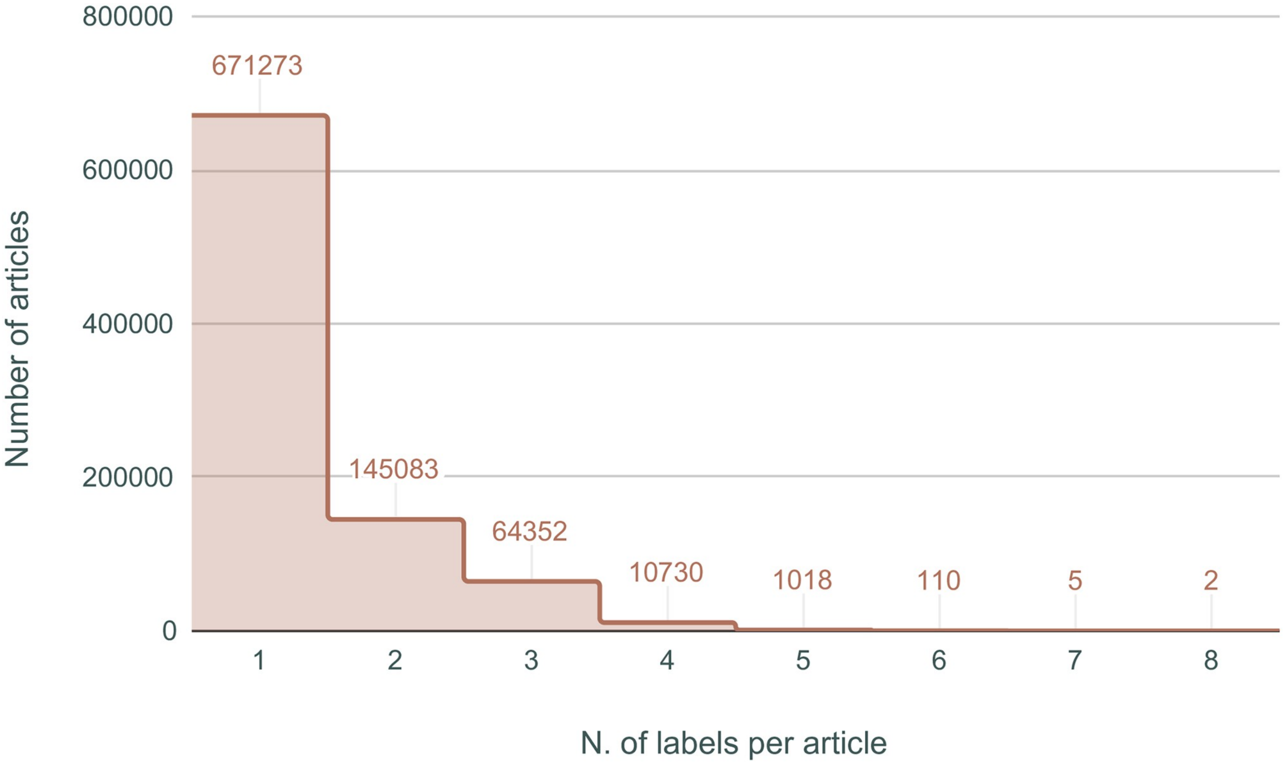

Fig 3. Distribution of the number of labels in ItWiki-100.

https://doi.org/10.1371/journal.pone.0270904.g003

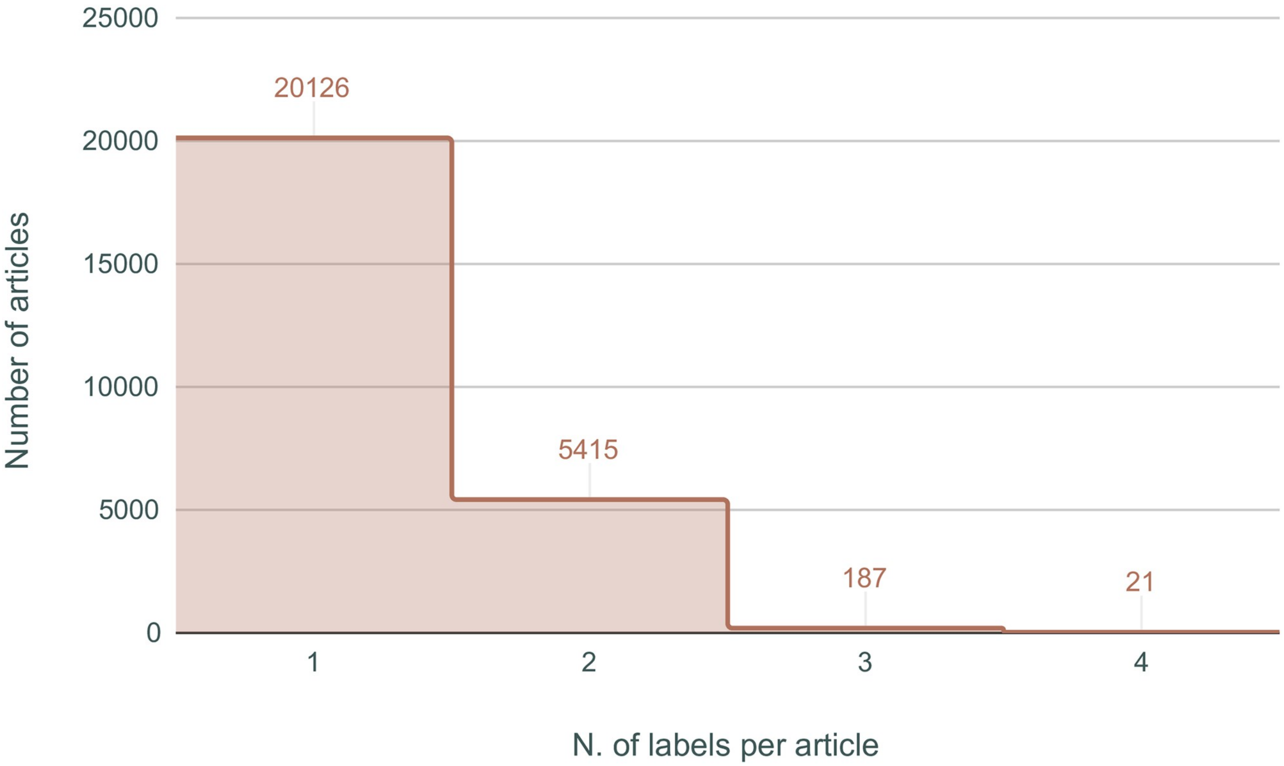

Fig 4. Distribution of the number of labels in RCV2it.

https://doi.org/10.1371/journal.pone.0270904.g004 processed, with a much smaller number of dominant classes and a much larger number of topics with next to no representation. As it stands, the dataset still remains unbalanced, but in a way that, in our opinion, poses an interesting challenge.

Experimental setup

For all methods, excluding FastText and classical approaches, the input documents are truncated, keeping only the first 512 tokens (or padded to that size). For training, every dataset is split into a training, validation, and test set: 40% of the data are reserved for testing, and 20% of the remaining samples are used for validation. Splits are produced in a way to preserve the distribution of labels, through a stratification strategy [250, 251]. Training was carried out on an NVIDIA GeForce RTX 2080 Ti GPU. More details on the training procedure are given in the S1 Appendix.

Evaluation metrics. One of the most adopted evaluation metrics for multilabel classification tasks is that of F1-score, defined as the harmonic mean of precision and recall, as in the equation below.

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

For multilabel and multiclass problems, it can be computed separately for each class and then averaged, obtaining the *macro* F1-score. In such a case, every class contributes equally to the final score, hence providing a more challenging metric for unbalanced datasets. On the other hand, a *micro* average reduction strategy is used when computing the metric globally with no weighting.

In our tests, we report the F1-score with macro-averaging across all categories along with the accuracy score, as the latter is an interpretable measure of the overall fraction of correct predictions. In its computation, the predicted set of labels must exactly match the ground truth for it to be considered a true positive (this score is sometimes referred to as "*subset accuracy*").

Methods applied. We present quantitative results for an array of models that either are or have been state-of-the-art approaches to solving the task of TC. A more thorough analysis of the decision process behind these models is provided in the S1 Appendix. We start by providing a strong baseline with classical methods, of which we test Naïve Bayes and linear Support Vector classifiers as representatives. As examples of neural networks preceding the Transformer era, we showcase the results of FastText [252], XML-CNN [241] and a BiLSTM-based classifier. We then trained Transformer-based methods, using open-sourced models pre-trained on language modeling tasks over large corpora. This last set of methods currently achieves the best results on the vast majority of downstream NLP tasks.

Every method is trained and tested 4 times per dataset, each time on a newly generated train and test split, and we list the final average metrics evaluated on the test set, along with the standard deviation over all runs, in Tables 10 and 11. As an exception, and because of technical constraints, we train XLM-R only once per dataset, as this model is much heavier than the already expensive BERT.

Discussion on results

When it comes to classical methods, the results obtained are quite favorable. The one-vs-rest ensemble of linear SVCs proved to be the strongest baseline, while our application of multinomial Naïve Bayes lagged behind quite a bit (though it was considerably faster). Among

Table 10. Test set macro F1 score for the tested TC methods.

	Italian		French		English	
Model	**ItWiki**	**RCV2it**	**FrWiki**	**RCV2fr**	**EnWiki**	**RCV1en**
Naïve Bayes (OVA)	0.620 ± 0.000	0.765 ± 0.004	0.551 ± 0.001	0.661 ± 0.003	0.636 ± 0.001	0.563 ± 0.014
Linear SVM (OVA)	0.824 ± 0.000	0.796 ± 0.008	0.737 ± 0.000	0.724 ± 0.003	0.803 ± 0.001	0.717 ± 0.003
FastText Classifier	0.815 ± 0.001	0.767 ± 0.007	0.757 ± 0.001	0.641 ± 0.007	0.744 ± 0.054	0.696 ± 0.008
BiLSTM (GloVe)	0.836 ± 0.001	0.805 ± 0.002	0.769 ± 0.001	0.700 ± 0.014	0.812 ± 0.005	0.766 ± 0.007
XML-CNN (FastText)	0.827 ± 0.001	0.776 ± 0.009	0.789 ± 0.002	0.669 ± 0.011	0.782 ± 0.004	0.730 ± 0.007
BERT (base)	**0.870** ± 0.001	**0.840** ± 0.006	**0.840** ± 0.001	**0.768** ± 0.005	**0.855** ± 0.002	**0.781** ± 0.004
XLM-R (base)	0.868	0.836	0.832	0.739	0.846	0.772

Standard deviation over runs is reported (± σ). Best results are presented in bold.

https://doi.org/10.1371/journal.pone.0270904.t010

Table 11. Test set subset accuracy score for the tested TC methods.

	Italian		French		English	
Model	**ItWiki**	**RCV2it**	**FrWiki**	**RCV2fr**	**EnWiki**	**RCV1en**
Naïve Bayes (OVA)	0.432 ± 0.001	0.629 ± 0.002	0.287 ± 0.001	0.475 ± 0.018	0.392 ± 0.007	0.473 ± 0.019
Linear SVM (OVA)	0.744 ± 0.001	0.717 ± 0.005	0.587 ± 0.001	0.656 ± 0.003	0.669 ± 0.001	0.677 ± 0.002
FastText Classifier	0.741 ± 0.002	0.678 ± 0.005	0.603 ± 0.001	0.611 ± 0.006	0.682 ± 0.053	0.670 ± 0.004
BiLSTM (GloVe)	0.763 ± 0.002	0.727 ± 0.007	0.637 ± 0.001	0.657 ± 0.008	0.680 ± 0.009	0.725 ± 0.005
XML-CNN (FastText)	0.764 ± 0.002	0.712 ± 0.005	0.661 ± 0.003	0.644 ± 0.008	0.666 ± 0.005	0.710 ± 0.001
BERT (base)	**0.808** ± 0.002	**0.773** ± 0.006	**0.724** ± 0.002	**0.696** ± 0.007	**0.753** ± 0.003	0.735 ± 0.002
XLM-R (base)	**0.808**	**0.773**	0.716	0.688	0.743	**0.740**

Standard deviation over runs is reported (± σ). Best results are presented in bold.

https://doi.org/10.1371/journal.pone.0270911

preprocessing operations, we found that lemmatization and *n*-gram discovery did not produce significant differences in results, so we do not report their effect in the final tables.

Neural networks that predate the Transformer era showcased strong performances, usually surpassing traditional methods. In our experiments, exceptions were likely to be attributed to the size of training data. NNs had better results for larger datasets, giving instead ground to classical methods whenever training samples were more scarce. Moreover, on smaller datasets (like RCV2it), we observed a noticeable margin of variance between the results of different runs of said networks, which were instead very consistent on larger datasets (like our "Wiki-100" datasets).

For these models, we experimented with different pre-trained embeddings (Word2Vec, GloVe, FastText), and found that FastText embeddings gave the best result for XML-CNN. In the case of BiLSTMs, however, we found the best results were instead obtained with GloVe embeddings, despite their comparatively restricted vocabulary size. Furthermore, BiLSTMs benefited from the removal of stopwords in their input text, something that we did not find to be true in the case of XML-CNN. Nonetheless, the gap in the results between the two models was noticeable but not dramatic.

Unsurprisingly, the attention-based Transformer architectures outshined other methods on all our datasets. An important aspect of these models that warrants being reiterated is their ease of application to downstream tasks. In fact, only a few epochs of tuning were necessary to obtain these results. Even so, they were still the most computationally complex and required the longest time to fine-tune. While monolingual BERT models performed best, XLM-R proved to have very strong performances, even though it is a multilingual model with a vocabulary diluted across many languages.

Language-specific considerations. On average, we observe that Italian models perform at the same level as (or slightly better than) English methods. French tasks, on the other hand, proved to be slightly harder on both datasets. In all cases, the trend of performance between methods is similar; as expected, contextualized language models perform the best across the line. We therefore focus this discussion on these models, are they are the most interesting to cover.

Many factors are likely to be influencing the differences in the reported results. First and foremost, the monolingual models were pre-trained on different corpora of different sizes. Diving into specifics, the Italian LM was pre-trained on 81GB and 13B tokens of data, taken from OPUS and OSCAR corpora [86]. The English BERT model is trained on the BookCorpus and Wikipedia dump (13GB), as outlined in the original paper [61, 245]. The French model is trained on a mixture of French documents, extracted from Project Gutenberg (a collection of e-books), OPUS, Wikimedia, and Common Crawl, amounting to 71 GB of data [245]. This latter model is also trained with a MLM objective only, while the others use both MLM and NSP, and it has more learnable parameters: 138 million instead of 110. Finally, XML-R was trained on 2.5TB of data in 100 languages, extracted from the Common Crawl corpus [70]. In it, the amount of data per language is variable: 300 GB for English, 30 GB for Italian, and 57 GB for French text. Another important factor to be considered is the difference in the size of our classification datasets. The RCV2 dataset contains a relatively small number of Italian articles when compared to both French and English. The number of target labels is also variable across languages, resulting in TC tasks that are likely to be on a slightly different level of difficulty.

As a consequence, given the conspicuous differences in pre-training, it is hard to make any consideration about the impact of the languages alone on the results. In this work, we aimed to give a generic overview from a linguistically inclusive perspective aimed at practical applications; indeed, we managed to obtain impressive results even without domain knowledge (for French) and without much fine-tuning. In future works, it would be certainly interesting to delve into a deeper study to ascertain the role of language and morphology in these models. Considering that tokenization is the most language-dependent step, this would involve testing several tokenizers, and pre-training the LMs from scratch on several monolingual corpora with adjusted language proportions, similarly to [7]. Clearly, this work would be very resource-intensive.

The similarity in results is not at all surprising, considering how close the three chosen languages are. Indeed, English has Germanic roots, while Italian and French are Romance languages (derived from Vulgar Latin), yet have developed very closely and have strong influences on each other. There are many differences that could be pointed out (gendered nouns and pronouns, liaisons, accents, etc.), but it is fair to consider them morphologically similar languages since they all belong to the fusional family.

Our results are suitable to prove the ease of application of pre-trained LMs and their convenience with respect to other traditional classification methods, as well as those based on LSTMs and MLPs with word embeddings. Despite our limited hyperparameter tuning imposed by a low-resource environment, these methods clearly show their value as one-and-for-all solutions for supervised TC. Moreover, the multilingual model XML-R was able to capture discriminative features in all three languages, in spite of the more limited per-language vocabulary.

Future research challenges

The last few years have seen exciting developments in the field of Text Classification and NLP in general. Large-scale language models have achieved state-of-the-art results throughout NLP literature, yet they are not infallible. These approaches face a set of challenges of their own regarding unexpected behaviors, true semantic understanding and harmful biases hidden in training data [3, 63]. Partially in response to these issues, new approaches are being researched, both to improve the reliability of LMs and to democratize their accessibility.

Multitask learning

The domain and language dependence of language models is one notable issue faced by LMs. Ideally, these models should show general understanding of languages via pre-training on several modeling tasks. We have mentioned how one of the limiting factors for fair experimentation of recent NLP models in languages other than English is the lack of downstream, task-specific datasets. We have shown such scarcity in Italian, which we expect only to be worse in lower-resourced languages. In this regard, multitask learning is a novel approach to learning language embeddings through combining labeled data from multiple related tasks and fine-tuning simultaneously on all of them, thus producing cross-task embeddings. Liu et al [253] proposed multitask DNN that showed strong generalization capabilities on domains where little-to-no labeled data was provided. They also provide evidence that this strategy profits from a regularization effect that reduces overfitting on single tasks, thus making embeddings more universal. The XGLUE [243] and XTREME [246] frameworks extend this concept by introducing a standard evaluation procedure for cross-task and cross-lingual models. It's easy to see how these paradigms could help tackle common problems for NLP research, first and foremost the scarcity of labeled, task-specific data for various languages.

Multilingual models

As shown, deep language models trained on large multilingual corpora are achieving excellent results [61, 70, 82], displaying a remarkable ability to extract semantic meaning across multiple languages. Given the computational complexity incurred by the development of Transformer-based models, it would perhaps be more desirable to push for the development of models that are able to generalize well to the widest array of languages possible. Multilingual models could help to prevent the newfound necessity of having to develop competitive monolingual language models for each language, which is becoming increasingly difficult due to the speed at which their dimension is growing. Furthermore, this would serve to benefit more under-represented languages, while partially addressing the justifiable ethical and environmental concerns related to the negative impact caused by the training process of these models [63]. Nonetheless, we have already addressed some of the limitations of these approaches, and how language-specific additions (such as a monolingual tokenizer) can improve performance.

New multilingual models are still being developed, often following the trend of what are colloquially defined as *large language models* (LLMs). Most of these projects are carried out by large tech companies, which are able to afford their vast computational costs; however, we point out notable scientific projects such as BigScience [254], currently being trained on 46 languages and more than 28 petaflops of textual data. It will be interesting to see the results of open projects such as these.

Few-shot learning

Another direction being researched is that of *few-shot learning*, where models are shown very little to no labeled examples in the fine-tuning procedure. Therefore, the aim is the creation of generic models able to overcome the lack of task-specific datasets; an example is that of the aforementioned GPT-3 [69], which was one of the first works in recent years to display impressive results without large amounts of task-specific data or model parameter updating.

Other works have followed, training on even larger datasets from diverse sources and experimenting with sparsely activated modules to address the computational expensiveness of LLMs [255–258].

An interesting aspect of these models is that they have shown to have "strong" multilingual capabilities. Many of these models indeed include corpora in different languages in their pretraining; however, while their results are certainly impressive, they are still outshined by monolingual approaches on language-specific tasks.

Reducing the size of language models

Though the trend of scaling larger and larger models still continues to this day, some developments are proposing smaller, generative LMs that have been shown to perform competitively when augmented with search/query information from retrieval databases [259, 260]. For instance, the developers of the Retrieval-Enhanced Transformer (RETRO) showcase performances on par with GPT-3, despite their model being 4% of the latter's size. As such, further research on the development of more reasonably sized models will be certainly a worthy endeavor.

Conclusion

In this paper, we overview existing models for TC and study their applicability to other languages, utilizing Italian as our main point of perspective. Firstly, we discuss the relevancy of preprocessing operations as arguably the most language-specific steps in the TC pipeline. We introduce the most common tokenization techniques that are paired with the latest methods and we further expand on different approaches to project textual features in suitable feature spaces for machine processing. Deep neural language models are introduced, and, whenever appropriate, we comment on the challenges and possible solutions to their applicability to non-English languages, first and foremost their high-resource requirements. A brief overview of the last step in the pipeline, that of classification, is then given; state-of-the-art approaches are outlined, commenting on their different levels of language dependence. We then showcase a number of Italian TC datasets, a language we deem mid-resourced; to substantiate this claim, we also similarly search for French datasets. We make a comparison between the two as well as with equivalent English datasets, showing that both French and English have greater availability of large labeled corpora. Furthermore, we give new quantitative results on multilabel classification tasks in Italian, French, and English. In particular, we apply a few main representatives of the methods we described on News Classification and Topic Labeling, two subcategories of TC which are underrepresented outside of the English scope. Finally, we discuss future research challenges and directions of TC, with an emphasis on how they affect other languages.

Supporting information

S1 Datasets. Charts with dataset statistics. Histograms with statistics about the RCV1/2 and Wiki datasets.
(PDF)

S1 Appendix. Training and testing procedure. Hyper-parameters used for training and relevant preprocessing operations.
(PDF)

Acknowledgments

We want to thank the Academic Editor and the reviewers for their support and precious advice.

Author Contributions

Data curation: Alessandro Zangari, Matteo Marcuzzo.

Formal analysis: Matteo Marcuzzo.

Investigation: Alessandro Zangari, Matteo Marcuzzo.

Methodology: Matteo Marcuzzo.

Project administration: Andrea Gasparetto.

Software: Alessandro Zangari, Matteo Marcuzzo.

Supervision: Andrea Gasparetto, Andrea Albarelli.

Validation: Andrea Gasparetto.

Writing – original draft: Andrea Gasparetto.

References

1. Bender EM. The #BenderRule: On Naming the Languages We Study and Why It Matters; 2019 Sep 14. In: The Gradient [Internet] [cited 2022 Apr 13]. Available from: https://thegradient.pub/the-benderrule-on-naming-the-languages-we-study-and-why-it-matters.
2. Magnini B, Cappelli A, Tamburini F, Bosco C, Mazzei A, Lombardo V, et al. Evaluation of Natural Language Tools for Italian: EVALITA 2007. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08); 2008 May 28–30. Marrakech, Morocco: European Language Resources Association (ELRA).
3. Bender EM. On Achieving and Evaluating Language-Independence in NLP. Linguistic Issues in Language Technology. 2011 Oct 01; 6. https://doi.org/10.33011/lilt.v6i.1239
4. Li Q, Peng H, Li J, Xia C, Yang R, Sun L, et al. A Survey on Text Classification: From Shallow to Deep Learning. arXiv. 2020 Aug 02;
5. Kowsari K, Jafari Meimandi K, Heidarysafa M, Mendu S, Barnes L, Brown D. Text Classification Algorithms: A Survey. Information. 2019 Apr 23; 10(4). https://doi.org/10.3390/info10040150
6. Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M, Gao J. Deep Learning–Based Text Classification: A Comprehensive Review. ACM Comput Surv. 2021 Apr; 54(3):1–40. https://doi.org/10.1145/3439726
7. Rust P, Pfeiffer J, Vulić I, Ruder S, Gurevych I. How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers); 2021 Aug. Online: Association for Computational Linguistics. p. 3118–3135.
8. Mielke SJ, Alyafeai Z, Salesky E, Raffel C, Dey M, Gallé M, et al. Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP. arXiv. 2021 Dec 20;abs/2112.10508.
9. Graves A. Generating Sequences With Recurrent Neural Networks. arXiv. 2013 Aug 4;abs/1308.0850.
10. Webster JJ, Kit C. Tokenization as the Initial Phase in NLP. In: Proceedings of the 14th Conference on Computational Linguistics - Volume 4. COLING'92; 1992 Aug 23–28. Nantes, France: Association for Computational Linguistics. p. 1106–1110.
11. Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, et al. Moses: Open Source Toolkit for Statistical Machine Translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions; 2007 Jun. Prague, Czech Republic: Association for Computational Linguistics. p. 177–180.
12. spacy.io [Internet]. Linguistic features—Tokenization; [cited 2022 Apr 13]. Available from:https://spacy.io/usage/linguistic-features##tokenization.

13. huggingface.co [Internet]. Tokenizer summary; [cited 2022 Apr 13]. Available from: https://huggingface.co/transformers/v3.0.2/tokenizer_summary.html.

14. Sennrich R, Haddow B, Birch A. 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 Aug 7–12. Berlin, Germany: Association for Computational Linguistics. p. 1715–1725.

15. Wang C, Cho K, Gu J. Neural Machine Translation with Byte-Level Subwords. vol. 34; 2020 Feb 7–12. New York, New York, USA. p. 9154–9160.

16. Schuster M, Nakajima K. Japanese and Korean voice search. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2012 Mar 25–30. Kyoto, Japan. p. 5149–5152.

17. Kudo T. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2018 Jul 15–20. Melbourne, Australia: Association for Computational Linguistics. p. 66–75.

18. Kudo T, Richardson J. 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 Oct 31—Nov 4. Brussels, Belgium: Association for Computational Linguistics. p. 66–71.

19. Gage P. A New Algorithm for Data Compression. The C Users J. 1994 Feb; 12(2):23–38.

20. Clément L, De la Clergerie E, Net L. MAF: a Morphosyntactic Annotation Framework. ResearchGate. 2005 Jan; Available from: https://www.researchgate.net/publication/228639144_MAF_a_Morphosyntactic_Annotation_Framework.

21. Stührenberg M. The TEI and Current Standards for Structuring Linguistic Data. Journal of the Text Encoding Initiative. 2012 Oct;3.

22. Rehman Z, Anwar W, Bajwa UI, Xuan W, Chaoying Z. Morpheme Matching Based Text Tokenization for a Scarce Resourced Language. PLOS ONE. 2013 Aug 21; 8(8):1–8. https://doi.org/10.1371/journal.pone.0068178 PMID: 23990871

23. Park D, Jang Y, Kim H. Korean-English Machine Translation with Multiple Tokenization Strategy. arXiv. 2021 May 29;abs/2105.14274.

24. Salameh M, Cherry C, Kondrak G. Reversing Morphological Tokenization in English-to-Arabic SMT. In: Proceedings of the 2013 NAACL HLT Student Research Workshop; 2013 Jun 9–14. Atlanta, Georgia, USA: Association for Computational Linguistics. p. 47–53.

25. Eifring H, Theil R. Linguistics for Students of Asian and African Languages. Universitetet i Oslo; 2005. Available from: https://www.uio.no/studier/emner/hf/ikos/EXFAC03-AAS/h05/larestoff/linguistics/Chapter%204.(H05).pdf.

26. Shao Y, Hardmeier C, Nivre J. Universal Word Segmentation: Implementation and Interpretation. Transactions of the Association for Computational Linguistics. 2018; 6:421–435. https://doi.org/10.1162/tacl_a_00033

27. Shapiro P, Duh K. BPE and CharCNNs for Translation of Morphology: A Cross-Lingual Comparison and Analysis. arXiv. 2018;abs/1809.01301.

28. Amrhein C, Sennrich R. How Suitable Are Subword Segmentation Strategies for Translating Non-Concatenative Morphology? In: Findings of the Association for Computational Linguistics: EMNLP 2021; 2021 Nov. Punta Cana, Dominican Republic: Association for Computational Linguistics. p. 689–705.

29. Salesky E, Etter D, Post M. Robust Open-Vocabulary Translation from Visual Text Representations. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing; 2021 Nov. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. p. 7235–7252.

30. Wu S, Dredze M. Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019 Nov 3–7. Hong Kong, China: Association for Computational Linguistics. p. 833–844.

31. Ács J. Exploring BERT's vocabulary; 2019 Feb 19. In: Judit Ács's blog [Internet] [cited 2022 Apr 13]. Available from: https://juditacs.github.io/2019/02/19/bert-tokenization-stats.html.

32. Ghag KV, Shah K. Comparative analysis of effect of stopwords removal on sentiment classification. In: 2015 International Conference on Computer, Communication and Control (IC4); 2015 Sep 10–12. Indore, India. p. 1–6.

33. Akhter MP, Jiangbin Z, Naqvi IR, Abdelmajeed M, Mehmood A, Sadiq MT. Document-Level Text Classification Using Single-Layer Multisize Filters Convolutional Neural Network. IEEE Access. 2020 Feb 27; 8:42689–42707. https://doi.org/10.1109/ACCESS.2020.2976744

34. Chen J, Huang H, Tian S, Qu Y. Feature selection for text classification with Naïve Bayes. Expert Systems with Applications. 2009 Apr;36(3, Part 1):5432–5435. https://doi.org/10.1016/j.eswa.2008.06.054

35. Mitra V, Wang CJ, Banerjee S. Text classification: A least square support vector machine approach. Applied Soft Computing. 2007 Jun; 7(3):908–914. https://doi.org/10.1016/j.asoc.2006.04.002

36. spacy.io [Internet]. Greek—SpaCy models; [cited 2022 Apr 13]. Available from: https://spacy.io/models/el.

37. spacy.io [Internet]. Italian—SpaCy models; [cited 2022 Apr 13]. Available from: https://spacy.io/models/it.

38. Jurafsky D, Martin JH. Speech and Language Processing. 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.; 2008 May 16.

39. Pistellato M, Bergamasco F, Albarelli A, Cosmo L, Gasparetto A, Torsello A. Robust phase unwrapping by probabilistic consensus. Optics and Lasers in Engineering. 2019 Oct; 121:428–440. https://doi.org/10.1016/j.optlaseng.2019.05.006

40. HaCohen-Kerner Y, Miller D, Yigal Y. The influence of preprocessing on text classification using a bag-of-words representation. PLOS ONE. 2020 May 1; 15(5):1–22. https://doi.org/10.1371/journal.pone.0232525 PMID: 32357164

41. Gasparetto A, Torsello A. A statistical model of Riemannian metric variation for deformable shape analysis. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015 Jun 7–12. IEEE. p. 1219–1228.

42. Jin P, Zhang Y, Chen X, Xia Y. Bag-of-Embeddings for Text Classification. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. IJCAI'16; 2016 Jul 9–15. New York, New York, USA: AAAI Press. p. 2824–2830.

43. O'Hara S, Draper BA. Introduction to the Bag of Features Paradigm for Image Classification and Retrieval. arXiv. 2011 Jan 17;abs/1101.3354.

44. Gasparetto A, Cosmo L, Rodola E, Bronstein M, Torsello A. Spatial Maps: From low rank spectral to sparse spatial functional representations; 2017 Oct 10–12. Qingdao, China: IEEE. p. 477–485.

45. Gasparetto A, Minello G, Torsello A. Non-parametric Spectral Model for Shape Retrieval. In: 2015 International Conference on 3D Vision; 2015 Oct 19–22. Lyon, France: IEEE. p. 344–352.

46. Jones KS. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation. 1972 Jan 1; 28(1):11–21. https://doi.org/10.1108/eb026526

47. Maćkiewicz A, Ratajczak W. Principal components analysis (PCA). Computers & Geosciences. 1993 Mar; 19(3):303–342. https://doi.org/10.1016/0098-3004(93)90090-R

48. Tharwat A, Gaber T, Ibrahim A, Hassanien AE. Linear discriminant analysis: A detailed tutorial. AI Communications. 2017 May 24; 30:169–190. https://doi.org/10.3233/AIC-170729

49. Tsuge S, Shishibori M, Kuroiwa S, Kita K. Dimensionality reduction using non-negative matrix factorization for information retrieval. In: 2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace. vol. 2; 2001 Oct 7–10. Tucson, Arizona, USA. p. 960–965.

50. Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space. arXiv. 2013 Jan 16;abs/1301.3781.

51. Pennington J, Socher R, Manning C. GloVe: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014 Oct 25–29. Doha, Qatar: Association for Computational Linguistics. p. 1532–1543.

52. Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics. 2017 Jun 1; 5:135–146. https://doi.org/10.1162/tacl_a_00051

53. Qin Q, Hu W, Liu B. Feature Projection for Improved Text Classification. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; 2020 Jul 5–10. Online: Association for Computational Linguistics. p. 8161–8171.

54. Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014 Oct 25–29. Doha, Qatar: Association for Computational Linguistics. p. 1724–1734.

55. Sutskever I, Vinyals O, Le QV. Sequence to Sequence Learning with Neural Networks. In: Proceedings of the 27th International Conference on Neural Information Processing Systems—Volume 2. NIPS'14; 2014 Dec 8–13. Cambridge, MA, USA: MIT Press. p. 3104–3112.

56. Smith NA. Contextual Word Representations: A Contextual Introduction. arXiv. 2019 Feb 15;

57. Xu H, Van Durme B, Murray K. BERT, mBERT, or BiBERT? A Study on Contextualized Embeddings for Neural Machine Translation. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing; 2021 Nov 7–11. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. p. 6663–6675.

58. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, et al. Deep Contextualized Word Representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers); 2018 Jun 1–6. New Orleans, Louisiana: Association for Computational Linguistics. p. 2227–2237.

59. Howard J, Ruder S. Universal Language Model Fine-tuning for Text Classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2018 Jul 15–20. Melbourne, Australia: Association for Computational Linguistics. p. 328–339.

60. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is All You Need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17; 2017 Dec 4–9. Long Beach, California, USA: Curran Associates Inc. p. 6000–6010.

61. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers); 2019 Jun 2–7. Minneapolis, Minnesota, USA: Association for Computational Linguistics. p. 4171–4186.

62. Radford A, Narasimhan K. Improving Language Understanding by Generative Pre-Training. 2018; Available from: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

63. Bender EM, Gebru T, McMillan-Major A, Shmitchell S. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency; 2021 Mar 3–10. Online, Canada: Association for Computing Machinery. p. 610–623.

64. Sevastjanova R, Kalouli AL, Beck C, Schäfer H, El-Assady M. Explaining Contextualization in Language Models using Visual Analytics. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers); 2021 Aug 1–6. Online: Association for Computational Linguistics. p. 464–476.

65. wikimedia.org [Internet]. Wikimedia Downloads; [cited 2022 Apr 13]. Available from: https://dumps.wikimedia.org/backup-index.html.

66. commoncrawl.org [Internet]. Common Crawl; [cited 2022 Apr 13]. Available from: https://commoncrawl.org.

67. Zhu Y, Kiros R, Zemel R, Salakhutdinov R, Urtasun R, Torralba A, et al. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In: 2015 IEEE International Conference on Computer Vision (ICCV); 2015 Dec 11–18. Santiago, Chile: IEEE Computer Society. p. 19–27.

68. Zhang J, Zhao Y, Saleh M, Liu P. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In: Proceedings of the 37th International Conference on Machine Learning. vol. 119 of ICML'20; 2020 Jul 13–18. Online: PMLR. p. 11328–11339.

69. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language Models are Few-Shot Learners. In: Advances in Neural Information Processing Systems 33 (NeurIPS 2020). vol. 33; 2020 Dec 6–12. Online: Curran Associates, Inc. p. 1877–1901.

70. Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, et al. Unsupervised Cross-lingual Representation Learning at Scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; 2020 Jul 5–10. Online: Association for Computational Linguistics. p. 8440–8451.

71. Li C. OpenAI's GPT-3 Language model: A technical overview; 2020 Jun 3. In: The Lambda Deep Learning Blog [Internet] [cited 2022 Apr 13]. Available from: https://lambdalabs.com/blog/demystifying-gpt-3.

72. Kaplan J, McCandlish S, Henighan T, Brown TB, Chess B, Child R, et al. Scaling Laws for Neural Language Models. arXiv. 2020;abs/2001.08361.

73. Lepikhin D, Lee H, Xu Y, Chen D, Firat O, Huang Y, et al. GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding. In: International Conference on Learning Representations (ICLR 2021); 2021 May 4. Vienna, Austria.

74. Fedus W, Zoph B, Shazeer N. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. arXiv. 2021;abs/2101.03961.

75. Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In: 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing —NeurIPS 2019; 2019 Dec 13. Vancouver BC, Canada: arXiv.org.

76. Jiao X, Yin Y, Shang L, Jiang X, Chen X, Li L, et al. TinyBERT: Distilling BERT for Natural Language Understanding. In: Findings of the Association for Computational Linguistics: EMNLP 2020; 2020 Nov 16–20. Online: Association for Computational Linguistics. p. 4163–4174.

77. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. arXiv. 2019;abs/1909.11942.

78. Clark K, Luong MT, Le QV, Manning CD. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In: ICLR 2020: Eighth International Conference on Learning Representations; 2020 Apr 26—May 1. Online.

79. Tamburini F. How "BERTology" Changed the State-of-the-Art also for Italian NLP. In: Proceedings of the Seventh Italian Conference on Computational Linguistics, CLiC-it 2020. vol. 2769 of CEUR Workshop Proceedings; 2021 Mar 1–3. Online.

80. Polignano M, Basile P, de Gemmis M, Semeraro G, Basile V. AlBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets. In: Proceedings of the Sixth Italian Conference on Computational Linguistics, CLiC-it 2019. vol. 2481 of CEUR Workshop Proceedings; 2019 Nov 13–15. Bari, Italy.

81. Mattei L, Cafagna M, Dell'Orletta F, Nissim M, Guerini M. GePpeTto Carves Italian into a Language Model. In: Proceedings of the Seventh Italian Conference on Computational Linguistics, CLiC-it 2020. vol. 2769; 2021 Mar 1–3. Online.

82. Pires T, Schlinger E, Garrette D. How Multilingual is Multilingual BERT? In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics; 2019 Jul 18—Aug 2. Florence, Italy: Association for Computational Linguistics. p. 4996–5001.

83. Pota M, Ventura M, Catelli R, Esposito M. An Effective BERT-Based Pipeline for Twitter Sentiment Analysis: A Case Study in Italian. Sensors. 2021; 21(1). https://doi.org/10.3390/s21010133

84. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. Transformers: State-of-the-Art Natural Language Processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; 2020 Nov 16–20. Online: Association for Computational Linguistics. p. 38–45.

85. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. TensorFlow: A System for Large-Scale Machine Learning. In: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation. OSDI'16; 2016 Nov 2–4. Savannah, GA, USA: USENIX Association. p. 265–283.

86. github.com [Internet]. Hugging Face + dbmdz Digital Library BERT models; [cited 2022 Apr 13]. Available from: https://github.com/dbmdz/berts.

87. github.com [Internet]. AlBERTo the first Italian BERT model for Twitter language understanding; [cited 2022 Apr 13]. Available from: https://github.com/marcopoli/AlBERTo-it.

88. github.com [Internet]. UmBERTo: an Italian Language Model trained with Whole Word Masking; [cited 2022 Apr 13]. Available from: https://github.com/musixmatchresearch/umberto.

89. github.com [Internet]. GilBERTo: An Italian pretrained language model based on RoBERTa; [cited 2022 Apr 13]. Available from: https://github.com/idb-ita/GilBERTo.

90. github.com [Internet]. GePpeTto GPT2 Model IT; [cited 2022 Apr 13]. Available from: https://github.com/LoreDema/GePpeTto.

91. de Vries W, Nissim M. As Good as New. How to Successfully Recycle English GPT-2 to Make Models for Other Languages. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021; 2021. Online: Association for Computational Linguistics. p. 836–846.

92. github.com [Internet]. GPT-2 Recycled for Italian and Dutch; [cited 2022 Apr 13]. Available from: https://github.com/wietsedv/gpt2-recycle.

93. huggingface.co [Internet]. BERT multilingual base model (cased); [cited 2022 Apr 13]. Available from: https://huggingface.co/bert-base-multilingual-cased.

94. huggingface.co [Internet]. XLM-RoBERTa (base-sized model); [cited 2022 Apr 13]. Available from: https://huggingface.co/xlm-roberta-base.

95. Rocchio JJ. Relevance feedback in information retrieval. In: The SMART Retrieval System: Experiments in Automatic Document Processing; 1971. NJ, USA. p. 313–323.

96. Xu S, Li Y, Wang Z. Bayesian Multinomial Naïve Bayes Classifier to Text Classification. In: Advanced Multimedia and Ubiquitous Engineering. FutureTech MUE 2017. Lecture Notes in Electrical Engineering. vol. LNEE 448; 2017 May 22–24. Seoul, Korea: Springer Singapore. p. 347–352.

97. Sutton C, McCallum A. An Introduction to Conditional Random Fields. Foundations and Trends® in Machine Learning. 2012 Aug 23; 4(4):267–373. https://doi.org/10.1561/2200000013

98. Bosch Avd. Hidden Markov Models. In: Encyclopedia of Machine Learning and Data Mining. Boston, MA: Springer US; Aug 2016. p. 1–3. Available from: https://doi.org/10.1007/978-1-4899-7502-7_124-1.

99. Cover T, Hart P. Nearest Neighbor pattern classification. IEEE Transactions on Information Theory. 1967 Jan; 13(1):21–27. https://doi.org/10.1109/TIT.1967.1053964

100. Cortes C, Vapnik V. Support-vector networks. Machine Learning. 1995 Sep; 20(3):273–297. https://doi.org/10.1007/BF00994018

101. Boser BE, Guyon IM, Vapnik VN. A Training Algorithm for Optimal Margin Classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory. COLT'92; 1992 Jul 27–29. New York, NY, USA: Association for Computing Machinery. p. 144–152.

102. Safavian SR, Landgrebe D. A survey of decision tree classifier methodology. IEEE Transactions on Systems, Man, and Cybernetics. 1991 May; 21(3):660–674. https://doi.org/10.1109/21.97458

103. Genkin A, Lewis D, Madigan D. Large-Scale Bayesian Logistic Regression for Text Categorization. Technometrics. 2007; 49. https://doi.org/10.1198/004017007000000245

104. Ho TK. Random decision forests. In: Proceedings of 3rd International Conference on Document Analysis and Recognition. vol. 1; 1995 Aug 14–16. Montreal, QC, Canada. p. 278–282.

105. Schapire RE. The Strength of Weak Learnability. Mach Learn. 1990; 5(2):197–227. https://doi.org/10.1023/A:1022648800760

106. Breiman L. Bagging Predictors. Machine Learning. 1996; 24(2):123–140. https://doi.org/10.1007/BF00058655

107. Iyyer M, Manjunatha V, Boyd-Graber J, Daumé III H. Deep Unordered Composition Rivals Syntactic Methods for Text Classification. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers); 2015 Jul 26–31. Beijing, China: Association for Computational Linguistics. p. 1681–1691.

108. Le Q, Mikolov T. Distributed Representations of Sentences and Documents. In: Proceedings of the 31st International Conference on Machine Learning. vol. 32 of Proceedings of Machine Learning Research; 2014 Jun 22-24. Beijing, China: PMLR. p. 1188–1196.

109. Tai KS, Socher R, Manning CD. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers); 2015 Jul 26–31. Beijing, China: Association for Computational Linguistics. p. 1556–1566.

110. Dieng AB, Wang C, Gao J, Paisley J. TopicRNN: A Recurrent Neural Network with Long-Range Semantic Dependency. In: 5th International Conference on Learning Representations, ICLR 2017, Workshop Track Proceedings; 2017 Apr 24–26. Toulon, France: OpenReview.net.

111. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Computation. 1997 Nov 15; 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 PMID: 9377276

112. Cho K, van Merriënboer B, Bahdanau D, Bengio Y. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In: Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation; 2014 Oct. Doha, Qatar: Association for Computational Linguistics. p. 103–111.

113. Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing. 1997 Nov; 45(11):2673–2681. https://doi.org/10.1109/78.650093

114. Pascanu R, Mikolov T, Bengio Y. On the Difficulty of Training Recurrent Neural Networks. In: Proceedings of the 30th International Conference on International Conference on Machine Learning—Volume 28. ICML'13; 2013 Jun 16–21. Atlanta, Georgia, USA: JMLR.org. p. III–1310–III–1318.

115. Zhang Y, Wallace B. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. In: Proceedings of the 8th International Joint Conference on Natural Language Processing (Volume 1: Long Papers); 2017 Nov 17—Dec 1. Taipei, Taiwan: Asian Federation of Natural Language Processing. p. 253–263.

116. Stone A, Wang H, Stark M, Liu Y, Phoenix D, George D. Teaching Compositionality to CNNs. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26. Honolulu, HI, USA: IEEE Computer Society. p. 732–741.

117. Pistellato M, Cosmo L, Bergamasco F, Gasparetto A, Albarelli A. Adaptive Albedo Compensation for Accurate Phase-Shift Coding. In: 2018 24th International Conference on Pattern Recognition (ICPR); 2018 Aug 20–24. Beijing, China: IEEE. p. 2450–2455.

118. Kim Y. Convolutional Neural Networks for Sentence Classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014 Oct 25–29. Doha, Qatar: Association for Computational Linguistics. p. 1746–1751.

119. Schiavinato M, Gasparetto A, Torsello A. Transitive Assignment Kernels for Structural Classification. In: Feragen A, Pelillo M, Loog M, editors. Similarity-Based Pattern Recognition; 2015 Oct 12–14. Copenhagen, Denmark: Springer International Publishing. p. 146–159.

120. Gasparetto A, Minello G, Torsello A. A Non-Parametric Spectral Model for Graph Classification. In: Proceedings of the International Conference on Pattern Recognition Applications and Methods—Volume 1; 2015 Jan 10–12. Lisbon, Portugal: SCITEPRESS—Science and Technology Publications, Lda. p. 312–319.

121. Cai H, Zheng VW, Chang K. A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications. IEEE Transactions on Knowledge & Data Engineering. 2018 Sep 1; 30(09):1616–1637. https://doi.org/10.1109/TKDE.2018.2807452

122. Bruna J, Zaremba W, Szlam A, Lecun Y. Spectral networks and locally connected networks on graphs. In: International Conference on Learning Representations (ICLR 2014); 2014 Apr 14–16. Banff, Canada.

123. Torsello A, Gasparetto A, Rossi L, Bai L, Hancock ER. Transitive State Alignment for the Quantum Jensen-Shannon Kernel. In: Fränti P, Brown G, Loog M, Escolano F, Pelillo M, editors. Structural, Syntactic, and Statistical Pattern Recognition; 2014 Aug 20-22. Joensuu, Finland: Springer Berlin Heidelberg. p. 22–31.

124. Yao L , Mao C , Luo Y . Graph Convolutional Networks for Text Classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33 of AAAI'19/IAAI'19/EAAI'19; 2019 Jan 27—Feb 1. Honolulu, Hawaii, USA: AAAI Press. p. 7370–7377.

125. Lin Y, Meng Y, Sun X, Han Q, Kuang K, Li J, et al. BertGCN: Transductive Text Classification by Combining GNN and BERT. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021; 2021 Aug 1–6. Online. p. 1456–1462.

126. Wu F, Souza A, Zhang T, Fifty C, Yu T, Weinberger K. Simplifying Graph Convolutional Networks. In: Chaudhuri K, Salakhutdinov R, editors. Proceedings of the 36th International Conference on Machine Learning. vol. 97 of Proceedings of Machine Learning Research; 2019 Jun 09–15. Long Beach, California, USA: PMLR. p. 6861–6871.

127. Li Q, Han Z, Wu XM. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence. No. 433 in AAAI'18/IAAI'18/EAAI'18; 2018 Feb 2–7. New Orleans, Louisiana, USA: AAAI Press. p. 3538–3545.

128. Gasparetto A, Marcuzzo M, Zangari A, Albarelli A. A Survey on Text Classification Algorithms: From Text to Predictions. Information. 2022 Feb 11; 13(2). https://doi.org/10.3390/info13020083

129. Liu PJ, Saleh M, Pot E, Goodrich B, Sepassi R, Kaiser L, et al. Generating Wikipedia by Summarizing Long Sequences;. Vancouver, BC, Canada.

130. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv. 2019 Jul 26;abs/1907.11692.

131. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language Models are Unsupervised Multitask Learners. 2019; Available from: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

132. Xue L, Barua A, Constant N, Al-Rfou R, Narang S, Kale M, et al. ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models. Transactions of the Association for Computational Linguistics. 2022; 10:291–306. https://doi.org/10.1162/tacl_a_00461

133. Wang A, Singh A, Michael J, Hill F, Levy O, Bowman S. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP; 2018 Nov. Brussels, Belgium: Association for Computational Linguistics. p. 353–355.

134. Basile P, Croce D, Basile V, Polignano M. In: Overview of the EVALITA 2018 Aspect-based Sentiment Analysis task (ABSITA); 2018. p. 10–16.

135. uniroma2.it [Internet]. ABSITA—Aspect-based Sentiment Analysis at EVALITA; [cited 2022 Apr 13]. Available from: http://sag.art.uniroma2.it/absita/data.

136. Barbieri F, Basile V, Croce D, Nissim M, Novielli N, Patti V. Overview of the Evalita 2016 SENTIment POLarity Classification Task. In: Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2016). vol. 1749 of CEUR Workshop Proceedings; 2016 Dec 5–7. Naples, Italy. p. 146–155.

137. european-language-grid.eu [Internet]. SENTIPOLC 2016 dataset; [cited 2022 Apr 13]. Available from: https://live.european-language-grid.eu/catalogue/corpus/7479.

138. De Mattei L, De Martino G, Iovine A, Miaschi A, Polignano M, Rambelli G. ATE ABSITA@ EVALITA2020: Overview of the Aspect Term Extraction and Aspect-based Sentiment Analysis Task. Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020), Online CEUR org. 2020;.

139. european-language-grid.eu [Internet]. ATE_ABSITA—Aspect Term Extraction and Aspect-based Sentiment Analysis Task; [cited 2022 Apr 13]. Available from: https://live.european-language-grid.eu/catalogue/corpus/7479.

140. european-language-grid.eu [Internet]. AMI 2020 Dataset; [cited 2022 Apr 13]. Available from: https://live.european-language-grid.eu/catalogue/corpus/7005.

141. Lai M, Cignarella AT, Hernández Farías DI, Bosco C, Patti V, Rosso P. Multilingual stance detection in social media political debates. Computer Speech & Language. 2020; 63:101075. https://doi.org/10.1016/j.csl.2020.101075

142. github.com [Internet]. mirkolai/MultilingualStanceDetection; [cited 2022 Apr 13]. Available from: https://github.com/mirkolai/MultilingualStanceDetection.

143. Viola L, Fiscarelli AM. From digitised sources to digital data: Behind the scenes of (critically) enriching a digital heritage collection. In: Proceedings of the International Conference Collect and Connect: Archives and Collections in a Digital Age. vol. 2810 of CEUR Workshop Proceedings; 2020 Nov. Online.

144. ChroniclItaly 3.0. A deep-learning, contextually enriched digital heritage collection of Italian immigrant newspapers published in the USA, 1898-1936; [cited 06.10.2021]. Available from: https://zenodo.org/record/4596345.

145. Sanguinetti M, Poletto F, Bosco C, Patti V, Stranisci M. An Italian Twitter Corpus of Hate Speech against Immigrants. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018); 2018 May 7–12. Miyazaki, Japan: European Language Resources Association (ELRA).

146. github.com [Internet]. Italian Hate Speech Corpus (IHSC); [cited 2022 Apr 13]. Available from: https://github.com/msang/hate-speech-corpus.

147. Sanguinetti M, Comandini G, Nuovo ED, Frenda S, Stranisci M, Bosco C, et al. HaSpeeDe 2 @ EVALITA2020: Overview of the EVALITA 2020 Hate Speech Detection Task. In: Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. vol. 2765 of CEUR Workshop Proceedings; 2020 Dec. Online.

148. unito.it [Internet]. HaSpeeDe2 Shared Task @ EVALITA 2020; [cited 2022 Apr 13]. Available from: http://www.di.unito.it/~tutreeb/haspeede-evalita20/data.html.

149. Croce D, Zelenanska A, Basili R. In: Neural Learning for Question Answering in Italian: XVIIth International Conference of the Italian Association for Artificial Intelligence; 2018. p. 389–402.

150. github.com [Internet]. SQuAD-it: A large scale dataset for Question Answering in Italian; [cited 2022 Apr 13]. Available from: https://github.com/crux82/squad-it.

151. Minard AL, Speranza M, Caselli T. The EVALITA 2016 Event Factuality Annotation Task (FactA). In: Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian; 2016 Dec 5–7. Naples, Italy.

152. fbk.eu [Internet]. Fact-Ita Bank; [cited 2022 Apr 13]. Available from: https://hlt-nlp.fbk.eu/technologies/fact-ita-bank.

153. Basili R, De Cao D, Lenci A, Moschitti A, Venturi G. EvalIta 2011: The Frame Labeling over Italian Texts Task. In: Magnini B, Cutugno F, Falcone M, Pianta E, editors. Evaluation of Natural Language and Speech Tools for Italian; 2013. Springer Berlin Heidelberg.

154. uniroma2.it [Internet]. EVALITA 2011 Frame Labeling over Italian Text; [cited 2022 Apr 13]. Available from: http://sag.art.uniroma2.it/flait/#resource.

155. Lyding V, Stemle E, Borghetti C, Brunello M, Castagnoli S, Dell'Orletta F, et al. The PAISÀ Corpus of Italian Web Texts. In: Proceedings of the 9th Web as Corpus Workshop (WaC-9); 2014 Apr. Gothenburg, Sweden: Association for Computational Linguistics. p. 36–43.

156. Lyding V, Stemle E, Borghetti C, Brunello M, Castagnoli S, Dell'Orletta F, et al.. PAISÀ Corpus of Italian Web Text; 2013. Available from: http://hdl.handle.net/20.500.12124/3.

157. Bosco C, Ballarè S, Cerruti M, Goria E, Mauri C. In: KIPoS @ EVALITA2020: Overview of the Task on KIParla Part of Speech Tagging. vol. 2765 of CEUR Workshop Proceedings; 2020 Jan.

158. github.com [Internet]. KIPOS2020; [cited 2022 Apr 13]. Available from: https://github.com/boscoc/kipos2020.

159. Basile P, Novielli N. In: Overview of the Evalita 2018 itaLIan Speech acT labEliNg (iLISTEN) Task. vol. 2263 of CEUR Workshop Proceedings; 2018 Jan. p. 44–50.

160. github.io [Internet]. iLISTEN, the first itaLIan Speech acT labEliNg task at Evalita 2018; [cited 2022 Apr 13]. Available from: https://ilisten2018.github.io/.

161. github.com [Internet]. PoSTWITA-UD; [cited 2022 Apr 13]. Available from: https://github.com/UniversalDependencies/UD_Italian-PoSTWITA.

162. Bosco C, Lombardo V, Vassallo D, Lesmo L. Building a Treebank for Italian: a Data-driven Annotation Schema. In: Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00); 2000 May. Athens, Greece: European Language Resources Association (ELRA).

163. unito.it [Internet]. Turin University Treebank; [cited 2022 Apr 13]. Available from: http://www.di.unito.it/~tutreeb/treebanks.html.

164. Bos J, Zanzotto FM, Pennacchiotti M. Textual Entailment at EVALITA 2009. 2009;.

165. evalita.it [Internet]. Textual Entailment EVALITA 2009 Dataset; [cited 2022 Apr 13]. Available from: http://www.evalita.it/2009/tasks/te.

166. Dell'Orletta F, Nissim M. In: Overview of the EVALITA 2018 Cross-Genre Gender Prediction (GxG) Task. vol. 2263 of CEUR Workshop Proceedings; 2018. p. 35–43.

167. sites.google.com [Internet]. Cross-Genre Gender Prediction in Italian; [cited 2022 Apr 13]. Available from: https://sites.google.com/view/gxg2018.

168. Menini S, Moretti G, Sprugnoli R, Tonelli S. DaDoEval @ EVALITA 2020: Same-Genre and Cross-Genre Dating of Historical Documents. In: Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. vol. 2765 of CEUR Workshop Proceedings; 2020. Online.

169. github.io [Internet]. Dating Document Evaluation at EVALITA 2020; [cited 2022 Apr 13]. Available from: https://dhfbk.github.io/DaDoEval/#data-and-annotation-description.

170. Brunato D, Chesi C, Dell'Orletta F, Montemagni S, Venturi G, Zamparelli R. AcCompl-it @ EVALITA2020: Overview of the Acceptability & Complexity Evaluation Task for Italian. In: Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. vol. 2765 of CEUR Workshop Proceedings; 2020. Online.

171. sites.google.com [Internet]. Acceptability & Complexity evaluation task for Italian at EVALITA 2020; [cited 2022 Apr 13]. Available from: https://sites.google.com/view/accompl-it/home-page.

172. Ronzano F, Barbieri F, Pamungkas EW, Patti V, Chiusaroli F. Overview of the EVALITA 2018 Italian Emoji Prediction (ITAMoji) Task. In: Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. vol. 2263 of CEUR Workshop Proceedings; 2018. Turin, Italy.

173. sites.google.com [Internet]. Data and Tools; [cited 2022 Apr 13]. Available from: https://sites.google.com/view/itamoji/data-and-tools.

174. kaggle.com [Internet]. French Twitter Sentiment Analysis; [cited 2022 Apr 13]. Available from: https://www.kaggle.com/hbaflast/french-twitter-sentiment-analysis.

175. Blard T. github.com [Internet]. French sentiment analysis with BERT; [cited 2022 Apr 13]. Available from: https://github.com/TheophileBlard/french-sentiment-analysis-with-bert.

176. Chiril P, Moriceau V, Benamara F, Mari A, Origgi G, Coulomb-Gully M. An Annotated Corpus for Sexism Detection in French Tweets. In: Proceedings of the 12th Language Resources and Evaluation Conference; 2020 May 11–16. Marseille, France: European Language Resources Association. p. 1397–1403.

177. github.com [Internet]. An Annotated Corpus for Sexism Detection in French Tweets; [cited 2022 Apr 13]. Available from: https://github.com/patriChiril/An-Annotated-Corpus-for-Sexism-Detection-in-French-Tweets.

178. Mazoyer B, Cagé J, Hervé N, Hudelot C. A French Corpus for Event Detection on Twitter. In: Proceedings of the 12th Language Resources and Evaluation Conference; 2020 May 11–16. Marseille, France: European Language Resources Association. p. 6220–6227.

179. Evrard M, Uro R, Hervé N, Mazoyer B. French Tweet Corpus for Automatic Stance Detection. In: Proceedings of the 12th Language Resources and Evaluation Conference; 2020 May 11–16. Marseille, France: European Language Resources Association. p. 6317–6322.

180. github.com [Internet]. Sentence embeddings for unsupervised event detection in the Twitter stream: study on English and French corpora; [cited 2022 Apr 13]. Available from: https://github.com/ina-foss/twembeddings.

181. Grabar N, Dalloux C, Claveau V. CAS: corpus of clinical cases in French. Journal of Biomedical Semantics. 2020 Aug 6; 11(1):7. https://doi.org/10.1186/s13326-020-00225-x PMID: 32762729

182. limsi.fr [Internet]. Défi Fouille de Textes@JEP-TALN 2020; [cited 2022 Apr 13]. Available from: https://deft.limsi.fr/2020/index-en.html.

183. d'Hoffschmidt M, Belblidia W, Heinrich Q, Brendlé T, Vidal M. FQuAD: French Question Answering Dataset. In: Findings of the Association for Computational Linguistics: EMNLP 2020; 2020 Nov 16–20. Online: Association for Computational Linguistics. p. 1193–1208.

184. european-language-grid.eu [Internet]. FQuAD: French Question Answering Dataset; [cited 2022 Apr 13]. Available from: https://live.european-language-grid.eu/catalogue/corpus/5007.

185. Keraron R, Lancrenon G, Bras M, Allary F, Moyse G, Scialom T, et al. Project PIAF: Building a Native French Question-Answering Dataset. In: Proceedings of the 12th Language Resources and Evaluation Conference; 2020 May 11–16. Marseille, France: European Language Resources Association. p. 5481–5490.

186. huggingface.co [Internet]. Datasets: piaf; [cited 2022 Apr 13]. Available from: https://huggingface.co/datasets/piaf.

187. elra.info [Internet]. Quaero Broadcast News Extended Named Entity corpus; [cited 2022 Apr 13]. Available from: http://catalog.elra.info/product_info.php?products_id=1195.

188. elra.info [Internet]. Quaero Old Press Extended Named Entity corpus; [cited 2022 Apr 13]. Available from: http://catalog.elra.info/product_info.php?products_id=1194.

189. elra.info [Internet]. A "scientific" corpus of modern French ("La Recherche" magazine) - Complete version; [cited 2022 Apr 13]. Available from: http://catalog.elra.info/product_info.php?products_id=595.

190. Abeillé A, Clément L, Liégeois L. Un corpus annoté pour le français: le French Treebank. Revue TAL. 2019; 60(2):19–43.

191. univ-paris-diderot.fr [Internet]. FTB: le French Treebank; [cited 2022 Apr 13]. Available from: http://ftb.linguist.univ-paris-diderot.fr/telecharger.php.

192. Leonhardt C, Blätte A. ParisParl Corpus of Parliamentary Debates [dataset]. 2020 May 10;

193. Laporte E, Nakamura T, Voyatzi S. A French Corpus Annotated for Multiword Nouns. In: Language Resources and Evaluation Conference. Workshop Towards a Shared Task on Multiword Expressions; 2008 Jun 1. Marrakech, Morocco. p. 27–30.

194. univ-mlv.fr [Internet]. French corpus annotated for multiword nouns; [cited 2022 Apr 13]. Available from: http://infolingu.univ-mlv.fr/english.

195. Amblard M, Beysson C, de Groote P, Guillaume B, Pogodalla S. A French Version of the FraCaS Test Suite. In: Proceedings of the 12th Language Resources and Evaluation Conference; 2020 May 11–16. Marseille, France: European Language Resources Association. p. 5887–5895.

196. inria.fr [Internet]. French Fracas Test Suite; [cited 2022 Apr 13]. Available from: https://gitlab.inria.fr/semagramme-public-projects/resources/french-fracas.

197. Prettenhofer P, Stein B. Cross-Language Text Classification Using Structural Correspondence Learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics; 2010 Jul. Uppsala, Sweden: Association for Computational Linguistics. p. 1118–1127.

198. Webis Cross-Lingual Sentiment Dataset 2010 (Webis-CLS-10); [cited 06.10.2021]. Available from: https://zenodo.org/record/3251672.

199. Keung P, Lu Y, Szarvas G, Smith NA. The Multilingual Amazon Reviews Corpus. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2020 Nov 16–20. Online: Association for Computational Linguistics. p. 4563–4568.

200. opendata.aws [Internet]. The Multilingual Amazon Reviews Corpus; [cited 2022 Apr 13]. Available from: https://registry.opendata.aws/amazon-reviews-ml/.

201. Pontiki M, Galanis D, Papageorgiou H, Androutsopoulos I, Manandhar S, AL-Smadi M, et al. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016); 2016 Jun. San Diego, California: Association for Computational Linguistics. p. 19–30.

202. qcri.org [Internet]. SemEval-2016 Task 5: Aspect-Based Sentiment Analysis; [cited 2022 Apr 13]. Available from: https://alt.qcri.org/semeval2016/task5/.

203. Lewis DD, Yang Y, Rose TG, Li F. RCV1: A New Benchmark Collection for Text Categorization Research. J Mach Learn Res. 2004; 5:361–397.

204. nist.gov [Internet]. Reuters Corpora (RCV1, RCV2, TRC2); [cited 2022 Apr 13]. Available from: https://trec.nist.gov/data/reuters/reuters.html.

205. Scialom T, Dray PA, Lamprier S, Piwowarski B, Staiano J. MLSUM: The Multilingual Summarization Corpus. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2020 Nov 16–20. Online: Association for Computational Linguistics. p. 8051–8067.

206. huggingface.co [Internet]. MLSUM Dataset; [cited 2022 Apr 13]. Available from: https://huggingface.co/datasets/mlsum.

207. kb.nl [Internet]. Europeana Newspapers NER; [cited 2022 Apr 13]. Available from: https://lab.kb.nl/dataset/europeana-newspapers-ner#access.

208. Pan X, Zhang B, May J, Nothman J, Knight K, Ji H. Cross-lingual Name Tagging and Linking for 282 Languages. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2017 Jul 30—Aug 4. Vancouver, Canada: Association for Computational Linguistics. p. 1946–1958.

209. european-language-grid.eu [Internet]. WikiAnn Dataset; [cited 2022 Apr 13]. Available from: https://live.european-language-grid.eu/catalogue/corpus/5192.

210. Spasojevic N, Bhargava P, Hu G. DAWT: Densely Annotated Wikipedia Texts Across Multiple Languages. In: Proceedings of the 26th International Conference on World Wide Web Companion. WWW'17 Companion; 2017 Apr 3–7. Perth, Australia: International World Wide Web Conferences Steering Committee. p. 1655–1662.

211. european-language-grid.eu [Internet]. Densely Annotated Wikipedia Texts (DAWT) Dataset; [cited 2022 Apr 13]. Available from: https://live.european-language-grid.eu/catalogue/corpus/4985.

212. Nothman J, Ringland N, Radford W, Murphy T, Curran JR. Learning Multilingual Named Entity Recognition from Wikipedia. Artif Intell. 2013; 194:151–175. https://doi.org/10.1016/j.artint.2012.03.006

213. metatext.io [Internet]. WikiNER Dataset; [cited 2022 Apr 13]. Available from: https://metatext.io/datasets/wikiner.

214. Minard AL, Speranza M, Urizar R, Altuna B, van Erp M, Schoen A, et al. MEANTIME, the NewsReader Multilingual Event and Time Corpus. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16); 2016 May. Portorož, Slovenia: European Language Resources Association (ELRA). p. 4417–4422.

215. newsreader-project.eu [Internet]. The NewsReader MEANTIME corpus; [cited 2022 Apr 13]. Available from: http://www.newsreader-project.eu/results/data/wikinews/.

216. Nivre J, de Marneffe MC, Ginter F, Goldberg Y, Hajič J, Manning CD, et al. Universal Dependencies v1: A Multilingual Treebank Collection. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16); 2016 May. Portorož, Slovenia: European Language Resources Association (ELRA). p. 1659–1666.

217. universaldependencies.org [Internet]. Universal Dependencies; [cited 2022 Apr 13]. Available from: https://universaldependencies.org/.

218. Raganato A, Pasini T, Camacho-Collados J, Pilehvar MT. XL-WiC: A Multilingual Benchmark for Evaluating Semantic Contextualization. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2020 Nov 16–20. Online: Association for Computational Linguistics.

219. github.io [Internet]. XL-WiC: The Multilingual Word-in-Context Dataset; [cited 2022 Apr 13]. Available from: https://pilehvar.github.io/xlwic/.

220. Benko V. Aranea: Yet Another Family of (Comparable) Web Corpora. In: Sojka P, Horák A, Kopeček I, Pala K, editors. Text, Speech and Dialogue; 2014. Springer, Cham: Springer International Publishing. p. 247–256. https://doi.org/10.1007/978-3-319-10816-2_31

221. uniba.sk [Internet]. Aranea: A Family of Comparable Gigaword Web Corpora; [cited 2022 Apr 13]. Available from: http://unesco.uniba.sk/aranea_about.

222. Toral A, Pecina P, Poch M, Way A. Towards a User-Friendly Webservice Architecture for Statistical Machine Translation in the PANACEA project. In: Proceedings of the 15th Annual conference of the European Association for Machine Translation; 2011 May 30–31. Leuven, Belgium: European Association for Machine Translation.

223. upf.edu [Internet]. PANACEA Project; [cited 2022 Apr 13]. Available from: http://lod.iula.upf.edu/resources/project_PANACEA#related-Corpus%20Text.

224. Longpre S, Lu Y, Daiber J. MKQA: A Linguistically Diverse Benchmark for Multilingual Open Domain Question Answering. Transactions of the Association for Computational Linguistics. 2021; 9:1389–1406. https://doi.org/10.1162/tacl_a_00433

225. github.com [Internet]. MKQA: Multilingual Knowledge Questions & Answers; [cited 2022 Apr 13]. Available from: https://github.com/apple/ml-mkqa/.

226. elra.info [Internet]. CLEF Question Answering Test Suites (2003-2008)—Evaluation Package; [cited 2022 Apr 13]. Available from: http://catalog.elra.info/en-us/repository/browse/ELRA-E0038/.

227. Conneau A, Rinott R, Lample G, Williams A, Bowman S, Schwenk H, et al. XNLI: Evaluating Cross-lingual Sentence Representations. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing; 2018 Oct–Nov. Brussels, Belgium: Association for Computational Linguistics. p. 2475–2485.

228. nyu.edu [Internet]. The Cross-Lingual NLI Corpus (XNLI); [cited 2022 Apr 13]. Available from: https://cims.nyu.edu/~sbowman/xnli/.

229. qwone.com [Internet]. The 20 Newsgroups data set; [cited 2022 Apr 13]. Available from: http://qwone.com/~jason/20Newsgroups.

230. uci.edu [Internet]. Reuters-21578 Text Categorization Collection; [cited 2022 Apr 13]. Available from: http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.

231. unipi.it [Internet]. AG's corpus of news articles; [cited 2022 Apr 13]. Available from: http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html.

232. Cer D, Yang Y, Kong Sy, Hua N, Limtiaco N, St John R, et al. Universal Sentence Encoder for English. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; 2018 Nov. Association for Computational Linguistics. p. 169–174.

233. Rajpurkar P, Jia R, Liang P. Know What You Don't Know: Unanswerable Questions for SQuAD. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers); 2018 Jul 15–20. Association for Computational Linguistics. p. 784–789.

234. Zhang X, Zhao J, LeCun Y. Character-Level Convolutional Networks for Text Classification. In: Proceedings of the 28th International Conference on Neural Information Processing Systems—Volume 1. NIPS'15; 2015 Dec. Cambridge, MA, USA: MIT Press. p. 649–657.

235. yelp.com [Internet]. Yelp Open Dataset: An all-purpose dataset for learning; [cited 2022 Apr 13]. Available from: https://www.yelp.com/dataset.

236. Maas AL, Daly RE, Pham PT, Huang D, Ng AY, Potts C. Learning Word Vectors for Sentiment Analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies; 2011 Jun. Portland, Oregon, USA: Association for Computational Linguistics. p. 142–150.

237. dbpedia.org [Internet]. DBpedia; [cited 2022 Apr 13]. Available from: https://www.dbpedia.org.

238. Williams A, Nangia N, Bowman S. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers); 2018 Jun 1–6. New Orleans, Louisiana: Association for Computational Linguistics. p. 1112–1122.

239. Tjong Kim Sang EF, De Meulder F. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003; 2003. p. 142–147.

240. Palmer A, Schneider N, Schluter N, Emerson G, Herbelot A, Zhu X, editors. Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021). Online: Association for Computational Linguistics; 2021 Aug.

241. Liu J, Chang WC, Wu Y, Yang Y. 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; 2017 Aug. Shinjuku, Tokyo, Japan: Association for Computing Machinery. p. 115–124.

242. Zhang W, Yan J, Wang X, Zha H. Deep Extreme Multi-Label Learning. In: Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval. ICMR'18; 2018 Jun. Yokohama, Japan: Association for Computing Machinery. p. 100–107.

243. Liang Y, Duan N, Gong Y, Wu N, Guo F, Qi W, et al. XGLUE: A New Benchmark Dataset for Cross-lingual Pre-training, Understanding and Generation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2020 Nov 16–20. Online: Association for Computational Linguistics. p. 6008–6018.

244. Wang A, Pruksachatkun Y, Nangia N, Singh A, Michael J, Hill F, et al. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R, editors. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. 294. Vancouver, Canada: Curran Associates Inc.; 2019 Dec 8–14. p. 3266–3280.

245. Le H, Vial L, Frej J, Segonne V, Coavoux M, Lecouteux B, et al. FlauBERT: Unsupervised Language Model Pre-training for French. In: Proceedings of the 12th Language Resources and Evaluation Conference; 2020 May 11–16. Marseille, France: European Language Resources Association. p. 2479–2490.

246. Hu J, Ruder S, Siddhant A, Neubig G, Firat O, Johnson M. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation. In: III HD, Singh A, editors. Proceedings of the 37th International Conference on Machine Learning. vol. 119 of Proceedings of Machine Learning Research; 2020 Jul 13–18. Online: PMLR. p. 4411–4421.

247. Sebastiani F. Machine Learning in Automated Text Categorization. ACM Comput Surv. 2002; 34(1):1–47. https://doi.org/10.1145/505282.505283

248. wikipedia.org [Internet]. Wikipedia:Portal; [cited 2022 Apr 13]. Available from: https://en.wikipedia.org/wiki/Wikipedia:Portal.

249. Dinu LP, Rusu A. Rank Distance Aggregation as a Fixed Classifier Combining Rule for Text Categorization. In: Proceedings of the 11th International Conference on Computational Linguistics and Intelligent Text Processing. CICLing'10; 2010 Mar 21–27. Iaşi, Romania: Springer-Verlag. p. 638—647.

250. Sechidis K, Tsoumakas G, Vlahavas I. On the Stratification of Multi-label Data. In: Gunopulos D, Hofmann T, Malerba D, Vazirgiannis M, editors. Machine Learning and Knowledge Discovery in Databases; 2011 Sep 5–9. Athens, Greece: Springer Berlin Heidelberg. p. 145–158.

251. Szymański P, Kajdanowicz T. A Network Perspective on Stratification of Multi-Label Data. In: Luís Torgo PB, Moniz N, editors. Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications. vol. 74 of Proceedings of Machine Learning Research; 2017 Sep 22. Skopje, Macedonia: PMLR. p. 22–35.

252. Joulin A, Grave E, Bojanowski P, Mikolov T. Bag of Tricks for Efficient Text Classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers; 2017 Apr 3–7. Valencia, Spain: Association for Computational Linguistics. p. 427–431.

253. Liu X, He P, Chen W, Gao J. Multi-Task Deep Neural Networks for Natural Language Understanding. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics; 2019 Jul 18—Aug 2. Florence, Italy: Association for Computational Linguistics. p. 4487–4496.

254. Sanh V, Webson A, Raffel C, Bach SH, Sutawika L, Alyafeai Z, et al. Multitask Prompted Training Enables Zero-Shot Task Generalization. arXiv. 2021;abs/2110.08207.

255. Du N, Huang Y, Dai AM, Tong S, Lepikhin D, Xu Y, et al. GLaM: Efficient Scaling of Language Models with Mixture-of-Experts. arXiv. 2021;abs/2112.06905.

256. Rae JW, Borgeaud S, Cai T, Millican K, Hoffmann J, Song HF, et al. Scaling Language Models: Methods, Analysis & Insights from Training Gopher. arXiv. 2021;abs/2112.11446.

257. Smith S, Patwary M, Norick B, LeGresley P, Rajbhandari S, Casper J, et al. Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model. arXiv. 2022;abs/2201.11990.

258. Chowdhery A, Narang S, Devlin J, Bosma M, Mishra G, Roberts A, et al. PaLM: Scaling Language Modeling with Pathways. arXiv. 2022;

259. Borgeaud S, Mensch A, Hoffmann J, Cai T, Rutherford E, Millican K, et al. Improving language models by retrieving from trillions of tokens. arXiv. 2021;abs/2112.04426.

260. Nakano R, Hilton J, Balaji S, Wu J, Ouyang L, Kim C, et al. WebGPT: Browser-assisted question-answering with human feedback. arXiv. 2021;abs/2112.09332.