Neurocomputing 470 (2022) 443–456

Contents lists available at ScienceDirect

Neurocomputing

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

An introduction to Deep Learning in Natural Language Processing: Models, techniques, and tools

Ivano Lauriola [a,b,1], Alberto Lavelli [c,*], Fabio Aiolli [b]

[a] Amazon Alexa AI
[b] University of Padova, Department of Mathematics, Italy
[c] Fondazione Bruno Kessler, Italy

ARTICLE INFO

Article history:
Received 6 January 2021
Revised 18 April 2021
Accepted 7 May 2021
Available online 22 July 2021

2020 MSC:
00-01
99-00

Keywords:
Deep Learning
Natural Language Processing
Transformer
Language Models
Software

ABSTRACT

Natural Language Processing (NLP) is a branch of artificial intelligence that involves the design and implementation of systems and algorithms able to interact through human language. Thanks to the recent advances of deep learning, NLP applications have received an unprecedented boost in performance. In this paper, we present a survey of the application of deep learning techniques in NLP, with a focus on the various tasks where deep learning is demonstrating stronger impact. Additionally, we explore, describe, and revise the main resources in NLP research, including software, hardware, and popular corpora. Finally, we emphasize the main limits of deep learning in NLP and current research directions.

1. NLP in a nutshell

Natural Language Processing (NLP) is a branch of artificial intelligence brimful of intricate, sophisticated, and challenging tasks related to the language, such as machine translation, question answering, summarization, and so on. NLP involves the design and implementation of models, systems, and algorithms to solve practical problems in understanding human languages.

We may split NLP into two main sub-branches, which are fundamental (or basic) and applicative research. Belonging to the first category, we find general problems representing the *bricks* to build complex systems based on human language. Some of these tasks are language modeling, morphological analysis, syntactic processing, or parsing, and semantic analysis. Additionally, NLP deals with applicative topics such as automatic extraction of relevant information (e.g., named entities and relations between them) from texts, translation of text between languages, summarization of documents, automatic answering of questions, classification and clustering of documents.

Thanks to the recent advances of deep learning, NLP applications have received an unprecedented boost in performance, generating growing interest from the Machine Learning community. For instance, in Machine Translation, the phrase-based statistical approaches that were at the state of the art have been gradually substituted with neural machine translation, consisting of huge deep neural networks that obtained better performance [1]. Similarly, early approaches for named entity recognition based on dictionaries, ontologies, and syntactic grammar rules have been replaced by recurrent architectures [2] and deep learning models. In both cases, large neural networks have demonstrated to be superior to traditional ML algorithms, such as SVM, for multiple reasons. Firstly, these models can often be trained with a single end-to-end architecture and they do not require traditional task-specific feature engineering, making their adoption convenient. Secondly, deep neural networks are able to handle a huge amount of training data. However, if we consider tasks related to the semantic analysis of natural languages, the limited availability of semantically annotated data, typically requiring specialized human effort, has slowed the diffusion of the neural approaches.

* Corresponding author.
E-mail addresses: lauivano@amazon.com (I. Lauriola), lavelli@fbk.eu (A. Lavelli), aiolli@math.unipd.it (F. Aiolli).

[1] The work of I.L. was done prior to joining Amazon Alexa AI.

https://doi.org/10.1016/j.neucom.2021.05.103
0925-2312/

Recent models also started to overtake the human performance on various tasks, such as Question Answering [3] or detection of deceiving contents [4].

However, even if recent techniques are starting to reach excellent performance on various tasks, there are still several problems that need to be solved, such as the computational cost, the reproducibility of results, and the lack of interpretability.

In the last years, various surveys concerning Deep Learning and Natural Language Processing have been published. Given the fast rate of progress in the field, we consider only the most recent surveys, i.e. those appeared since 2019 [5–9].

Our paper, rather than a survey, aims at being a tutorial for the Machine Learning community. In this perspective, it provides (i) a classification of the main NLP tasks; (ii) an analysis of current issues and future work, focusing on reproducibility; (iii) a description of software and hardware resources and of the main corpora used in NLP. Some remarks concerning the previous surveys. Specifically, the authors in [5] survey different architectures of the classic neural network language models and their improvements. [9] reviews deep learning models from the perspective of text representation learning. [8] provides a brief introduction to both NLP and deep neural networks, and discusses how deep learning is being used to solve current problems in NLP. [7] categorizes and addresses the different aspects and applications of NLP that have benefited from deep learning. Finally, the survey in [6] concentrates on cross-lingual word embedding models.

2. Tasks and applications

Due to the ubiquitous human–computer interaction, NLP techniques are currently used in several different tasks, covering multiple domains. Most of modern NLP applications can be categorized in the following classes:

Sequence classification. These NLP problems are full-fledged classification tasks. Let $\mathscr{X}$ be a set of input sequences, where each sequence $s \in \mathscr{X}$ is a series of tokens $s = \langle w_1 \dots w_{|s|} \rangle$ and let $\mathscr{Y}_c = \{c_1, c_2 \dots\}$ be a set of possible classes. Similarly to common classification problems in Machine Learning, the aim of sequence classification is to find a function $f_c : \mathscr{X} \rightarrow \mathscr{Y}_c$ able to assign a class to each sequence. Some relevant examples are (i) sentiment analysis, whose purpose is to classify a short text according to its polarity, (ii) document categorization, that finds the topic of a document (e.g., sport, finance...), and (iii) answer sentence selection, where the goal is to select the best sentence from a given paragraph/text to answer an input question.

Pairwise sequence classification. Pairwise sequence classification consists in comparing and classifying two different sequences according to their similarity, their semantics, and their meanings. Usually, pairwise sequence classification is a binary classification task where, given two different sequences as input, returns +1 iff they express the same meaning, -1 otherwise, that is $f_p : \mathscr{X} \times \mathscr{X} \rightarrow \{+1, -1\}$. Algorithms and models for these tasks need to fully understand a sequence and to extract meaningful semantic representations, overcoming multiple problems like synonymy and polysemy. One of the most popular applications is the Quora Question Pairs challenge,[2] whose aim is to find duplicated questions from Quora.

Word labeling. In word labeling applications, a label is attached to each token $w_i \in s$. Specifically, the output space $\mathscr{Y}_w$ consists of sequences of labels for each element of the input $y = \langle y_1 \dots y_{|s|} \rangle \in \mathscr{Y}_w$. Examples of word labeling tasks are (i) Named Entity Recognition (NER), where relevant entities (e.g., names, locations) are identified from the input sequence, (ii) classical question answering, where a probability distribution issued by an input paragraph is used to select a span containing the answer, or (iii) Part-of-Speech (PoS) tagging, that is the process of marking up a word in a text as corresponding to a particular part of speech (verb, noun, adjective, ...).

Sequence2sequence. In seq2seq problems, the input sequence is used to generate an output sequence. Differently from word labeling applications, the input sequence s and the output sequence y are not directly aligned, i.e. $|s| \neg = |y|$, and the model needs to *generate* a new sentence. Although either the input $\mathscr{X}$ and the output $\mathscr{Y}_{s2s}$ spaces contain sequences, they may be disjoint sets, as is the case of machine translation.

Note that this classification is not exhaustive but covers the most popular and relevant tasks. An example of a few NLP tasks applied to the input sentence "*I really like David's cat!*" is shown in Fig. 1.

3. Recent advances in NLP

One of the main issues in the last decade for NLP applications was the definition of a suitable and effective representation of tokens, sentences, and documents. Early approaches described a word w_i from a given dictionary Σ as one-hot encoding $\boldsymbol{h}_{w_i} \in \{0,1\}^{|\Sigma|}$ (see Fig. 2). This solution has two main drawbacks. Firstly, input words are described by huge vectors whose dimension depends on the dictionary size. Secondly, different words have orthogonal representations $\boldsymbol{h}_{w_i} \perp \boldsymbol{h}_{w_j}$, with a consequent drop of any possible semantic relations between words. This aspect has strongly limited the capability of NLP systems, unable for instance to catch the similarities between *apple*, *kiwi*, *table*, *peach* words and discover unrelated word.

3.1. Word and sentence vectors

Recently, Mikolov et al. [10] proposed an efficient and effective method to learn distributed low-dimensional word-level representations, known as *word vectors* or *word embeddings*, such that words with similar meaning have similar representations. The method, named Word2vec, consists of a shallow neural network with an encoder-decoder structure pre-trained on unlabeled corpora. Similarly to an autoencoder, the network tries to reconstruct a neighbor word (context) w_j given an input *target* word w_i, that is $\boldsymbol{h}_{w_i} \overset{enc}{\rightarrow} \boldsymbol{v}_{w_i} \overset{dec}{\rightarrow} \boldsymbol{h}_{w_j}$, where $\boldsymbol{v}_{w_i} \in \mathbb{R}^d$ is the word embedding of w_i. Two different models, CBOW and Skip-gram, have been proposed. The former is trained to reconstruct a target word given its context as input, whereas the latter tries to predict context words given the target word. Word2vec has also shown its capability to capture a large number of precise syntactic and semantic word relationships. For example, the analogy *"king is to queen as man is to woman"* is encoded in the resulting vector space as the equation $\boldsymbol{v}_{king} - \boldsymbol{v}_{queen} = \boldsymbol{v}_{man} - \boldsymbol{v}_{woman}$ (see Fig. 3).

Due to its effectiveness on several tasks, such as NER [11], sentiment analysis [12], recommendation [13], and synonym recognition [14], Word2vec received considerable attention in the literature, and several improved solutions have subsequently been proposed. Some relevant examples are (i) Global-Vector (GloVe) [15], that exploits statistical information computed on the whole corpus, and (ii) fastText [16], that injects sub-words (character n-grams) information to describe the inner structure of a word. This inner structure can be extremely useful in several applications, such as Biomedical text mining [17], where for instance affixes of biomedical terms have a specific structure.

[2] https://www.kaggle.com/c/quora-question-pairs.

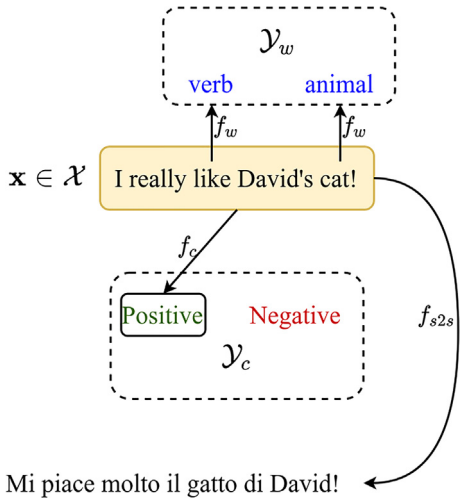


Fig. 1. Examples of NLP tasks applied to the same input sentence, including NER, PoS, sentiment analysis, and machine translation.

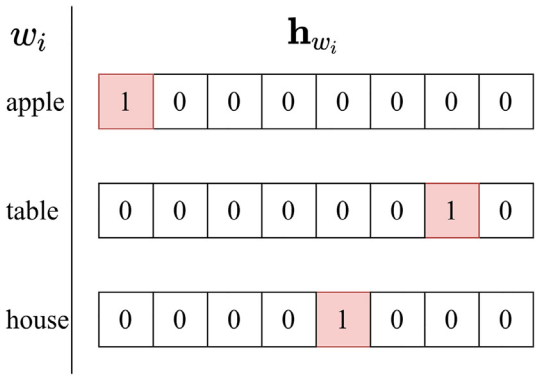


Fig. 2. Example of one-hot encoding.

Beyond the models presented in this section, which are the most popular and relevant, the literature on word embeddings covers more than a decade of research that cannot be easily summarized in this type of paper. Countless surveys exist on this topic. For instance, Almeida et al. [18] covers the evolution of word embeddings starting from the first neural models proposed in 2003, emphasizing the differences between them and the intuition behind the models. Differently, focusing on cross-lingual setting, the survey provided by Ruder et al. [6] represents a strong contribution.

However, despite the impressive results of word vectors, the definition of a suitable representation for sentences and texts is still challenging. One of the main approaches commonly used for this purpose predates the explosion of deep learning and is known as Bag-of-Words (BOW) [19]. BOW represents a document d as its (countable) set of words that compose it, and it can be computed as the sum of one-hot word vectors that compose the document $\sum_{w_i \in d} \boldsymbol{h}_{w_i}$. This approach is really intuitive and the resulting feature vector is able to describe the content of a document. However, the dimension of the feature vector quickly increases with the dictionary size, and the semantic of the text is not taken into account. BOW representations have been widely used in the literature, such as in spam filtering [20] and document classification [21,22]. With the advent of word vectors, new methods to develop meaningful document and sentence level representations have been proposed. These methods can be categorized into two classes, i.e. unsupervised document embedding techniques, typically inspired by

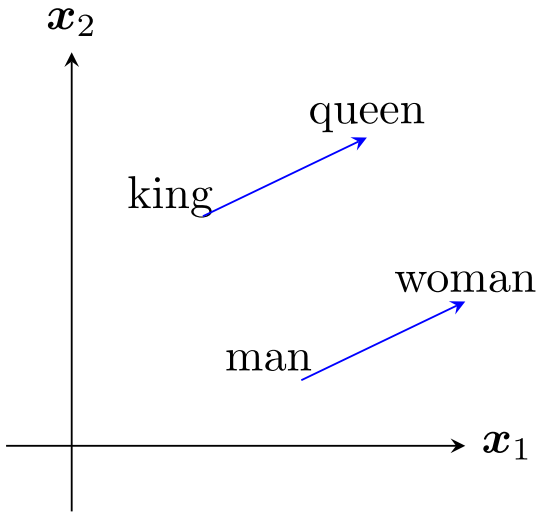


Fig. 3. Example of relation learned by Word2Vec.

Word2vec, and supervised approaches. Unsupervised word/sentence vectors aim at extracting general representations that can be placed in various tasks. These methods can be trained on large scale unlabeled corpora through a language model objective function, which is a probability distribution over sequences of words. On the other hand, supervised methods use explicit labels to develop meaningful representations used in downstream tasks.

As a primer attempt of unsupervised method, the simple average pooling of word vectors has been explored to derive sentence vectors [23]. Consecutively, different methods that directly extend Word2vec have been released, as is the case of Doc2Vec (also known as ParagraphVector) [24]. A further relevant solution was Skip-thought vectors [25] that is based on the same structure of skip-gram, but it replaces the atomic units from words to sentences. Given a target sentence, Skip-thought tries to reconstruct a context sentence. An encoder-decoder structure based on RNN with GRU units is involved. Other newsworthy approaches are fastSent [26], which extends Skip-thought vectors, and [27], that uses a combination of CNN (encoder) and RNN (decoder).

Several methods have also been proposed in supervised scenarios. Most of them are based on recursive [28], recurrent, or convolutional neural networks [29]. Usually, these methods build a neural network on the top of word vectors, combining the properties of pre-trained word embeddings, the elasticity of neural architectures, and the strength of the supervision.

For instance, the neural network for sequence text classification proposed in [30] takes a static word vector for each input word and then it combines them through a CNN layer with multiple filters and feature maps (see Fig. 4). The authors also shows significant improvements when the word vectors are initialyzed with pre-trained embeddings fine-tuned on the targe task. A similar approach has been used in [31], where a siamese CNN is employed to rank short text pairs.

One relevant application of such technologies is neural Machine Translation (MT), where sequence2sequence neural networks have been proposed as encoder-decoder (one for each language) architecture [32,33]. Unlike the previous phrase-based translation system [34] that consists of many small sub-components separately tuned, neural MT tries to build a single but larger neural network that reads a sentence as input and returns the translation as output. The main issue with this approach is that the information coming from long sentences cannot be compressed in a fixed-length vector (see Fig. 5), named *context* vector, with a consequent drop in performance. To this end, attention mechanisms [1] have

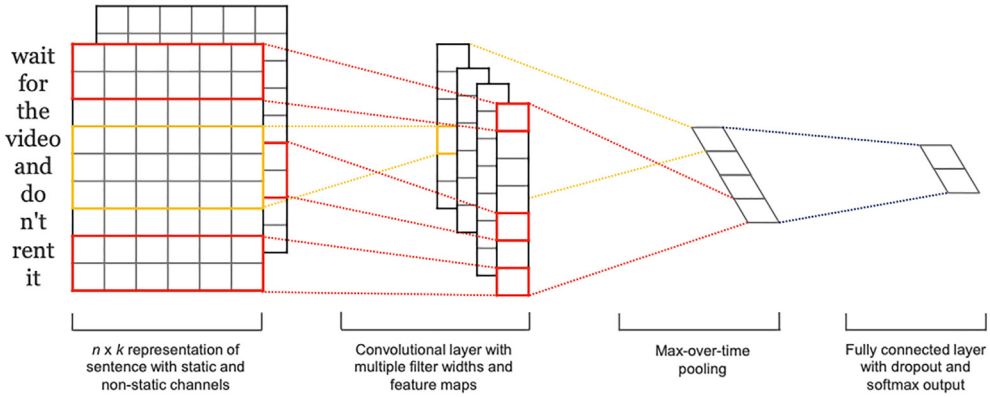


Fig. 4. The typical structure of a CNN for sequence classification [30]. Static word vectors are used as input. Then, convolutional layers learns semantic relations between words.

been introduced, where the context vector used to produce each output state is defined as a linear combination of all internal encoding contexts (Fig. 6). The model showed remarkable results when dealing with long sentences.

Inspired by the recent success of bidirectional RNN [35,36], ELMo [37] (Embeddings from Language Models) is probably one of the most interesting methods emerging from a plethora of works and previous attempts. In short, instead of using a static word vector, ELMo looks at the entire sentence producing a contextualized word embedding through a bidirectional language model. The network is a multilayer LSTM (see Fig. 7) pre-trained on unlabeled data. Most important, the authors showed mechanisms to use internal representations in downstream tasks by fine-tuning the network, improving results on several benchmarks.

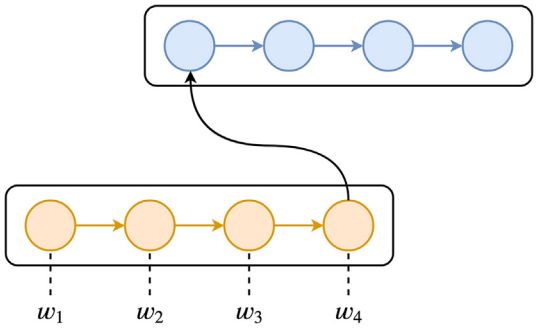


Fig. 5. Classical sequence2sequence architecture based on recurrent neural networks. The encoder (orange) produces a context vector to fed the decoder (blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.2. Pre-trained Transformer models

However, the last real boost in NLP after the advent of word vectors and unsupervised pre-training is the Transformer model [38]. The Transformer is the first architecture entirely based on attention to draw global dependencies between input and output, replacing the recurrent layers most commonly used in encoder-decoder architectures. The model showed a new state of the art in translation quality, while it can be trained significantly faster than architectures based on recurrent or convolutional layers. The evolution of language models pre-trained on large unlabeled corpora and the surprisingly empirical effectiveness of Transformer architectures are the two main pillars of modern NLP. One of the most popular pre-trained Transformer models is BERT [39] (Bidirectional Encoder Representations from Transformers). BERT is designed to pre-train deep bidirectional representations from unlabeled texts by jointly conditioning on both left and right contexts in all layers. The pre-training was driven by two language model objectives, i.e. Masked Language Model (MLM) and Next Sentence Prediction (NSP). In MLM, showed in Fig. 8, the network masks a small number of words of the input sequence and it tries to predict them in output, whereas in NSP the network tries to understand the relations between sentences by means of a binary loss. Specifically, the model has to select if two sentences are consecutive or not. After a pre-training phase, the model can be easily used in downstream tasks by fine-tuning the network on the target domain. BERT can be used in several different tasks, such as sequence classification, word- labeling, sequence2sequence, and so on. These methods rely on two main strengths, (i) the architecture strongly based on self-attention mechanisms that allow to read and to keep track of the whole input sequence, and (ii) the

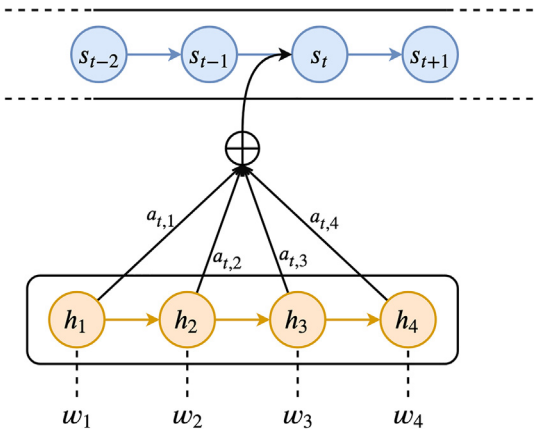


Fig. 6. The attention mechanism allows to produce an output state by means of a combination of intermediate context vectors.

pre-training that allows the network to read and to (at least apparently) understand a text, its semantic and the meaning.

Inspired by BERT, several pre-trained Transformers have been subsequently proposed, as is the case of RoBERTa [40], ALBERT [41], and DistilBERT [42]. These extensions of BERT were based on the same Transformer architecture with few small differences, without introducing additional features. For instance, RoBERTa criticized the NSP loss arguing that NSP is a critical task also for

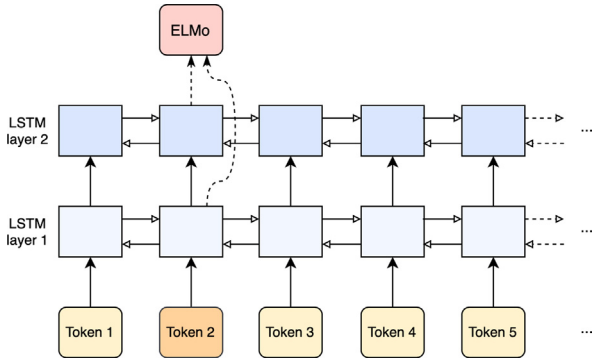


Fig. 7. ELMoo: the word embedding assigned to each token is a function of the whole sentence. Note that the intermediate outputs from the two LSTM layers are both used to define the final representation.

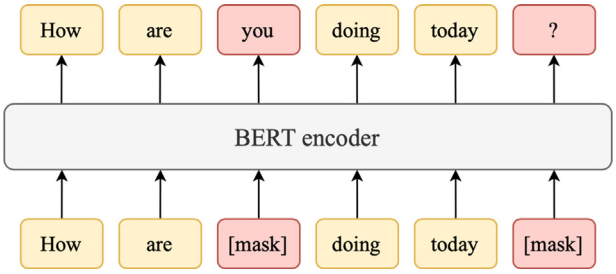


Fig. 8. Masked Language Model.

humans, and it does not improve the performance of the network. Other relevant methods based on the same concepts are GPT [43,44] (Generative Pre-Training), Transformer-XL [45], and its extension XLNet [46].

Nowadays, these methods are continuously achieving excellent performance on a plethora of NLP tasks, such as question answering [47–49], text classification [50], sentiment analysis [51], biomedical text mining [52], and Named Entity Recognition [53]. Surprisingly, these networks started to overcome human performance on several tasks that were considered unsolvable by AI, such as Question Answering [3] and verbal lie detection [4].

And on top of that, pre-trained Transformers have shown impressive performance in cross multilingual scenarios. For instance, Pires et al. [54] showed that Multilingual-BERT (M-BERT) pre-trained on 104 languages is extremely robust to develop cross-language representations without using an explicit multilingual training objective. Other authors [55] emphasized the fact that the lexical overlap between languages plays a negligible role in the cross-lingual success.

Transformer-based architectures have also been widely used for text generation, i.e. the task of generating a text (typically a paragraph of a sentence) given an input passage, including summarization, translation, or chatbot. A popular resource in this field is GPT (Generative Pre-Training) [43]. GPT uses 12-layer decoder only transformer structure with masked self-attention to train language model on 7000 unpublished books. Released in 2019, GPT-2 [44] extends its predecessor GPT. With a few minor changes, the network consists of a 24-layers Transformer with 1.5 billions of learnable parameters. The context size as been also increased from 512 to 1024 tokens. Most important, GPT-2 introduced the concept of *Task Conditioning*, for which the training allows to learn multiple tasks using the same unsupervised model (that is: $P(output|input, task)$). In other words, the model is expected to produce different output for same input for different tasks. The task conditioning for language models is performed by providing examples or natural language instructions to the model to perform a task. Task conditioning forms the basis for zero-shot task transfer.

The recently proposed GPT-3 [56], an autoregressive language model with 175 billion parameters, further improves the performance of GPT-2, showing impressive results in text generation. The authors showed the ability of the model in generating samples of news articles which human evaluators have difficulty distinguishing from articles written by humans. Notably, GPT-3 has shown effective in zero- and few-shot settings, resizing the importance of fine-tuning for very large language models. However, GPT-3 still has several drawbacks, including its drop of coherence when generating long texts and the high cost of inference. Additionally, GPT-3 raises some ethical and sociological problems, including the risk of using its generation capability phishing, spamming, spreading misinformation, or performing other fraudulent activities.

While their main success was in Natural Language Generation, impressive results were also achieved for other tasks.

4. Resources

In this section, we describe the main resources in NLP research and development, including software and scientific libraries, corpora, and hardware analysis for running large-scale state-of-the-art models, focusing on Transformers.

4.1. Software

NLP attracted, in the past decades, a consistent number of developers and scientists who have made available a plethora of libraries, tools, and scripts to handle with both low-level NLP modules (tokenization, PoS tagging...) and high-level systems (document classifiers, models...).

In the following we provide a brief description of the main tools that we consider most relevant for performing NLP tasks. The selection of these tools is driven by (i) the set of functionalities that they provide, (ii) the dimension of the community behind the resource, and (iii) the usability.

NLTK [57][3] (Natural Language Toolkit) is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

Gensim [58][4] is a Python library for topic modelling, document indexing and similarity retrieval with large corpora. Target audience is NLP and Information Retrieval (IR) communities. The library contains efficient implementations of popular algorithms, such as Latent Semantic Analysis (LSA/LSI/SVD), Latent Dirichlet Allocation (LDA), Random Projections (RP), Hierarchical Dirichlet Process (HDP). The library also includes several Word2vec pre-trained models.

SpaCy [59][5] is an industrial-strength library for performing NLP tasks in Python. The library is specifically designed to build complex industrial systems, and it interoperates seamlessly with TensorFlow, PyTorch, scikit-learn, Gensim and the rest of Python's AI ecosystem. Spacy includes several functionalities, such as tokenization, NER, sentence segmentation, PoS tagging, and depen-

[3] https://www.nltk.org/.
[4] https://radimrehurek.com/gensim/.
[5] https://spacy.io/.

dency parsing. The library also contains various pre-trained word vectors.

Transformers [60][6] provides general-purpose architectures (BERT, GPT-2, RoBERTa, XLNet...) for Natural Language Understanding (NLU) and Natural Language Generation (NLG) with over 32+ pretrained models in 100+ languages and deep interoperability between TensorFlow 2.0 and PyTorch. The library is curated by *Huggingface*, an NLP-focused startup with a large open-source community. The library exposes APIs to use many well-known pre-trained transformer architectures described in the previous sections.

CoreNLP [61][7] is a NLP library encompassing the main NLP modules and tasks, i.e. tokenization, PoS tagging, parsing, NER, for the Java programming language. The library provides support for 6 different languages.

Stanza [62][8] is a collection of accurate and efficient tools for many human languages. Stanza includes a Python interface to the CoreNLP Java package. The toolkit is designed to be parallel among more than 70 languages, using the Universal Dependencies formalism. Finally, the package provides specialized tools and models for Biomedical and clinical tasks.

The main characteristics and details of these libraries are summarized in Table 1.

4.2. Hardware

The second important ingredient in NLP research is the hardware to run experiments and to train large-scale deep neural networks, such as the Transformer. In this section we will show an overview of the main hardware resources for running NLP models surrounded by thoughts, tips, and technical details.

Usually, small research labs use their own machines with a couple of affordable GPUs, such as the Nvidia RTX 2080-Ti or 1080-Ti. Notwithstanding these solutions are not expensive[9] and they do not require specialized cooling systems, they have some drawbacks. Firstly, these GPUs are not designed for running 24/7 applications. Secondly, they have a limited amount of memory (about 11 GiB for a 2080/1080-Ti), which is undersized for recent models. Differently, medium and large research labs leverage different and optimized solutions, such as High Performance Computing (HPC) clusters, as is the case of our lab, or on-demand web-services, such as Amazon Web Services (AWS) or Microsoft Azure, which provide their optimized GPUs for a discrete hourly cost.

As previously discussed, NLP research is brimful of different tasks with different characteristics, complexities, and needs. Hence, the selection of the correct hardware is fundamental in order to maximize the productivity and to limit the overall cost, which includes the initial hardware price, the maintenance, and the energy consumption. This last aspect may be relevant as most of GPUs for deep learning have a Thermal Design Power (TDP) of 250–400 Watt.

There are several key criteria in selecting the most suitable GPU for a given purpose, and they include (i) the training speed (influenced by the number of CUDA cores, the architecture, the frequency...), (ii) the amount of memory on which the maximum batch size depends, (iii) the efficiency, i.e. the TDP, (iv) the system that manages the GPU (blade, workstation, desktop), and clearly (v) the cost of the GPU.

Having said that, we evaluated a subsection of popular GPUs showing their performance in relation to their cost and other parameters when dealing with Transformer models. Our assessed hardware comprises:

Nvidia Tesla V100: The V100 is, probably, the most known and the fastest GPU for deep learning in 2017–2020. It is paired with 16 (or 32) GiB of memory, allowing us to operate with discrete batch sizes.

Nvidia Tesla A100: This card has been introduced in the middle of 2020, significantly overcoming the V100. The A100 is paired with 40 GiB of memory (Nvidia announced a version with 80 GiB) and it is becoming the new standard for large-scale deep learning applications.

Nvidia Tesla T4: The T4 is not extremely fast, but it is specifically designed for low power consumption (only 75 W TDP). Thanks to this characteristic, it became popular in large HPCs.

Nvidia Tesla K80: The K80 is a popular solution widely used in the past. Each card contains 2 GPU chips with 11 GiB. We included this card in our analysis as it is extremely cheap at the moment of writing (a K80 card, i.e. 2 GPUs, can be purchased for 150–200€) and, notwithstanding it is designed with a passive cooling, it can be effectively installed in a desktop.

Nvidia Tesla M60: As is the case of the K80, a M60 card consists of 2 individual GPUs of 8 GiB each. Unfortunately, its price is significantly higher than the price of the K80. Moreover, the limited amount of memory (8×2) may limit its potentialities with recent models.

Intel Xeon E5-2686-v4 (i): this is a popular CPU widely used for these applications. It includes 18 physical cores and 36 threads at 2.3 Ghz (base clock). It supports AVX-2 instructions for fast vectorial operations.

The characteristics of these GPUs are summarized in Table 2.

We empirically compared these GPUs on a simple fine-tuning task. Specifically, we extracted a subset of sequences from GLUE, a popular benchmark dataset described in the next sections and we fine-tuned a BERT-base-uncased model on those sequences. We used a batch size of 16 to allow a fair comparison on all GPUs. We also applied dynamic padding, where the overall length of a batch is defined by the longest sequence that comprises it. The average batch length is 148. Finally, we collected the average number of sentences that the networks (and the CPU) can process in a second. Results of this comparison are shown in Fig. 9. The experiment ran on a Linux system with CUDA 11.0, PyTorch 1.7, and Transformers 3.5 installed.

Beyond the mere training speed, selecting the right fleet is not trivial. For instance, old Nvidia Tesla cards, such as the K80 or M40, are much slower than a recent V/A100, but they are extremely cheap nowadays and they are paired with 24 GB of memory. As a further example, let us compare the Nvidia Tesla K80 and the Tesla T4, which are two GPUs with similar speed (if you consider the two chips inside the K80 card together). On the one hand, the K80 is extremely cheaper compared to the T4 (200$ against 2,000$) and it has much more memory (12×2 against 16). On the other hand, the latter is much more efficient, allowing to regain the initial cost in long-term scenarios.

The energy consumption should not be underestimated for two main reasons, which are the energy cost and the consequently carbon footprint [63,64]. Recent findings in NLP literature show that the computations required for deep learning research have been doubling every few months, resulting in an estimated 300,000x increase from 2012 to 2018 [65]. In order to better explore this aspect, Fig. 10 shows the speed of fine-tuning BERT-base compared to the TDP.

However, there are other aspects that characterize a GPU. For instance, recent Nvidia GPUs (Turing, Volta, and Ampere) are highly optimized for mixed precision computation with FP32 and

[6] https://huggingface.co/transformers/.
[7] https://stanfordnlp.github.io/CoreNLP/.
[8] https://stanfordnlp.github.io/stanza/.
[9] At the moment of writing, a 2080-Ti can be purchased for 600–650€.

Table 1
NLP software libraries and their characteristics.

Library	1st release	Language	Multilingual	License
NLTK	2001	python	yes	Apache 2.0
Gensim	2009	python	partially	GNU LGPLv2+
SpaCy	2015	python	yes	MIT license
Transformers	2019	python	partially	Apache 2.0
CoreNLP	2010	Java	yes	GNU GPLv3
Stanza	2019	Python	yes	Apache 2.0

Table 2
Main characteristics of popular hardware widely used in deep learning and NLP.

Model	Year	Memory GiB	TFlops FP32	TDP Watt	Price €
V100 SXM2	2017	16	15.67	250	2600–3000
A100 SXM4	2020	40	19.49	400	4000–10000
T4	2018	16	8.14	75	1500–2000
K80	2014	12 × 2	4.11 × 2	150 × 2	150–200
M60	2015	8 × 2	4.83 × 2	150 × 2	1600–2000
E5-2686-v4	2016	–	0.6	145	400

FP16 operations, improving the scalability of large-scale models without affecting the final result.[10] Also, GPUs equipped with Tensor cores (as is the case of the T4) may further accelerate typical deep learning workloads such as feed-forward and convolutional layers when using mixed-precision. A final mention in this overview of hardware is reserved to high-performance Application Specific Integrated Circuit (ASIC) chips that are designed to accelerate machine learning workloads during inference time. These chips include, for instance, Tensor Processing Unit (TPU) or AWS Inferentia.

4.3. Datasets

Starting from the 1990's, NLP has been characterized by increasing efforts in organizing competitions and shared tasks (see [66] for a description of early efforts), where multiple teams are challenged to solve specific problems with standard datasets and public leaderboards. Additionally, thanks to the recent explosion of deep learning and the abundance of increasingly intelligent models, NLP tasks quickly evolved growing in complexity. A primer example is provided by Question Answering (QA): a task that were considered, in the past, extremely complex for a machine is now partially solved as state-of-the-art models and industrial virtual assistants are ready to answer complex open-domain questions in a multi-turn conversation.

In the remainder of this section we briefly describe the reference datasets of the most relevant NLP tasks, with a particular attention for those more suitable for the DL approaches.

4.3.1. Question Answering

As we already introduced in this work, QA is the task of producing an answer given the question and a passage (or short text). There are mainly two distinct tasks in QA literature. The first is the Answer Sentence Selection (AS2), and it consists of selecting the sentence constituting (or containing) the answer. The second task, that is known as Machine Reading (MR), consists of identifying the exact text span representing the answer.

First QA models were based on linguistic theory, and they leverage the concept of words overlapping (or Redundancy-based techniques [67]), assuming that the passage containing the answer partially overlap with the question (e.g., the passage "*Abraham Lincoln was an American statesman and lawyer...*" cannot answer the question "*Who is Lady Gaga?*" as these two sentences do not share any entity). Surprisingly, some authors showed that simple word matching outperformed various sophisticated systems [68]. With the advent of neural networks, several models belonging to the *compare-aggregate* framework have been proposed [69,70]. In short, these networks act in two phases. Firstly, they try to develop a contextualized representation for the question and candidate answers (e.g., sentences from a text). Then, they compare these representations selecting the candidate answer that better represents the question. Thanks to the ability of these new models in developing a suitable neural representation, QA gained attraction and corpora moved from simple and factoid questions towards open domain QA.

A popular and widely used open domain MR datasets is the Stanford Question Answering Dataset (SQuAD-1.1) [71]. The cor-

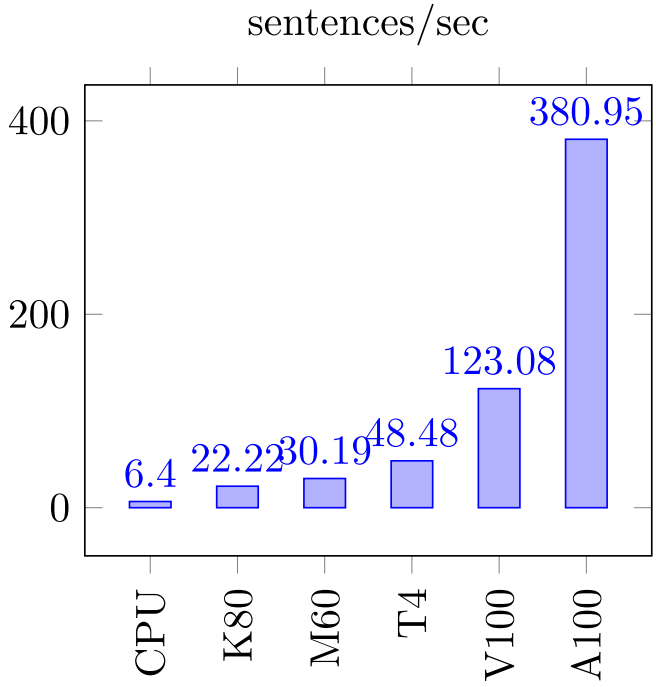


Fig. 9. Hardware speed when fine-tuning BERT-base-uncased pre-trained model. In the case of dual GPUs (K80 and M60), a single chip is considered.

[10] https://github.com/huggingface/transformers/tree/master/examples/text-classification#mixed-precision-training.

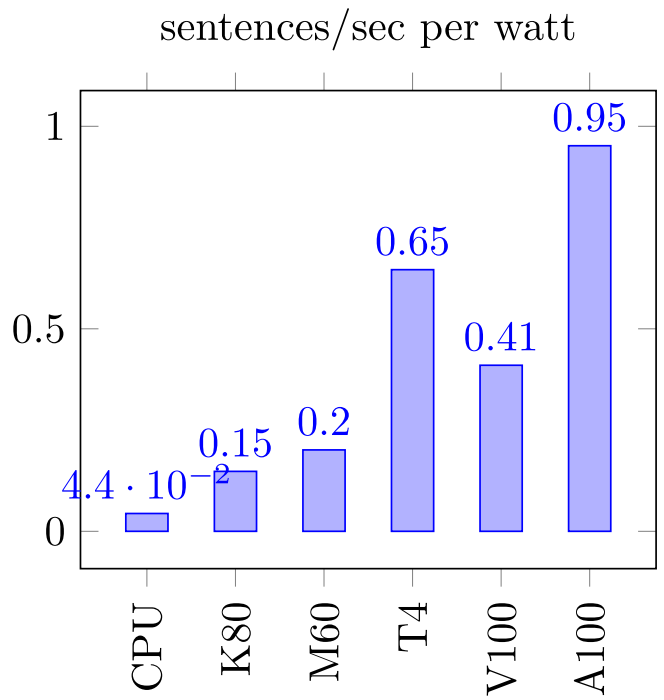


Fig. 10. Hardware speed per watt when fine-tuning BERT-base-uncased pre-trained model.

pus consists of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage. Recently, an evolution of SQuAD-1.1, namely SQuAD-2.0 [72], has been introduced. SQuAD-2.0 combines the 100,000 questions in SQuAD-1.1 with other 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones. To do well on SQuAD-2.0, systems must not only answer questions, but also determine when no answer is supported by the paragraph and abstain from answering. This last sub-task is known as answer triggering. The main drawback of SQuAD is that the passage containing the question consists of a simple short paragraph, and recent systems overcome the human performance.[11]

Thanks to the recent breakthrough carried by the Transformer network, QA is evolving towards three main directions. Firstly, inspired by practical reasons (e.g., virtual assistants), various large corpora consisting of complex user-generated questions and long passages (e.g., complete webpages and not a simple paragraph) have been proposed. One of the most important resources for MR in this scenario is Natural Questions (NQ) [73]. NQ contains real user questions issued to Google search, and answers found from Wikipedia by annotators. The resource comprises 307,373 training examples. Differently from SQuAD or other previous corpora, like WikiQA [74], another small but widely used resource for AS2, the text containing (or not) the answer consists of a complete Wikipedia article retrieved by a search engine rather than a simple paragraph, making the task more challenging. An adaptation of NQ for AS2, named ASNQ, has been recently proposed [48].

The second research direction concerns the interaction (or dialog) between a questioner, who is trying to learn about a text, and an answerer. In this scenario, two fundamental resources are Question Answering in Context (QuAC) [75] and Conversational Question Answering Challenge (CoQA) [76]. Both of them are surrounded by a discrete community providing scripts, code, and baselines.

The third direction focuses on the question and its understandability. Is the question comprehensible? Does the question contain an ambiguity? Does it require a clarification? Recently, large-scale corpora have been released to train models for these purposes, such as [77,78].

Another relevant resources for AS2 that deserve a mention in this summary is MS-Marco [79], which contains 100,000 open-domain questions and an associated text retrieved from Wikipedia.

4.3.2. Sentiment Analysis

Sentiment Analysis (SA) is the task of identifying the people's feeling about a specific event or entity. One of the most popular applications of SA in the NLP field concerns the product reviews, where the goal is to predict the user's feeling about a purchased item or a watched movie. This information is extremely helpful for companies that aim at improving their systems (e.g., e-commerce) by analyzing the users' ratings and feelings. Usually, SA is a sequence classification task where, given the input text, the system returns the label representing the feeling.

One of the most popular SA datasets widely used to benchmark NLP models is the IMDB Movie Reviews Dataset [80]. The dataset consists of 50,000 movie reviews of which 25,000 are labelled as positive intent and 25,000 as negative.

Much effort has been also devoted in collecting data from Twitter. The most popular large-scale SA corpus generated from Twitter traffic is, without a doubt, Sentiment140 [81]. The corpus consists of 1,400,000 tweets, and it is used to analyze user responses to different products, brands, or topics through user tweets. However, the corpus has been collected with a distant supervised approach based on the emoticons inside the tweets to define the label. For instance, a tweet containing the emoticon ":)" is classified as positive. Clearly, this approach does not take various problems into account, such as the irony and the sarcasm.

Moving towards different domains, the Paper Reviews Data Set [82] contains 405 reviews of scientific papers, and they are labelled with a score from −2 (very negative) to 2 (very positive).

As is the case of other tasks, SA has recently evolved towards the combination of information from multiple sources, e.g. text, images, and videos. Some relevant corpora for *Multimodal* SA are (i) IEMOCAP [83], a multimodal and multi-speaker database containing 12 h of audiovisual data, the labels are categorical and reflect different emotional states, such as anger, happiness, and sadness; (ii) MOSI [84], an opinion-level annotated corpus of sentiment and subjectivity analysis in online videos composed by 3,702 distinct examples; and (ii) its extension CMU-MOSEI [85], which contains 23,453 distinct examples.

4.3.3. Machine Translation

A comprehensive resource for MT is OPUS,[12] a collection of translated texts from the web. The OPUS project aims to convert and align free online data, to add linguistic annotation, and to provide the community with a publicly available parallel corpus. OPUS is based on open source products and the corpus is also delivered as an open content package. All pre-processing is done automatically. No manual corrections have been carried out. OPUS provides a wide coverage of MT resources but it lacks specific indications about the relevance of the single resources.

The reference source for the use of MT datasets is WMT,[13] the most important MT evaluation campaign. On its website, it is possible to find the parallel corpora used in various evaluation campaigns. The main corpora are Europarl [86] and News Commentary [87], and UN [87] for some language pairs.

[11] https://rajpurkar.github.io/SQuAD-explorer/.

[12] http://opus.nlpl.eu/.

[13] http://www.statmt.org/wmt20/.

To train a good general-purpose model it is necessary to exploit parallel texts of at least hundreds of millions words. To obtain such amount of words, ParaCrawl[14] and OpenSubtitles [88][15] can be used. Another interesting resource is MultiUN [87], a collection of translated documents from the United Nations. For translations from/to English, there is OPUS-100,[16] an English-centric multilingual corpus covering 100 languages.

It is worthwhile also to mention *WIT*^{3} [89] (Web Inventory of Transcribed and Translated Talks). *WIT*^{3} is a ready-to-use version for research purposes of the multilingual transcriptions of TED talks.[17] Since 2007, the TED Conference has been posting on its website all video recordings of its talks, English subtitles and their translations in more than one hundred languages. In order to make this collection of talks more effectively usable by the research community, the original textual contents are redistributed here, together with MT benchmarks and processing tools.

As for speech translation, there are the following two resources: MuST-C [90] and Europarl-ST [91]. MuST-C is a multilingual speech translation corpus whose size and quality facilitates the training of end-to-end systems for speech translation from English into several languages. For each target language, MuST-C comprises several hundred hours of audio recordings from English TED Talks, which are automatically aligned at the sentence level with their manual transcriptions and translations. Europarl-ST is a Multilingual Speech Translation Corpus, that contains paired audio-text samples for Speech Translation, constructed using the debates carried out in the European Parliament in the period between 2008 and 2012.

4.3.4. Biomedical text mining

With the enormous volume of biological literature and its rapid growth, Biomedical text mining is becoming increasingly important to provide a structured and rapid access to actionable Biomedical information. Formally, Biomedical text mining (BioNLP) refers to the methods and study of how text mining and NLP techniques may be applied to the Biomedical literature and texts.

Recently, deep learning has boosted the development of effective biomedical text mining models. However, directly applying NLP techniques and models to biomedical text mining often yields unsatisfactory results due to a word distribution shift from general domain corpora to biomedical corpora [52]. To this end, several NLP tasks (and resources) have been specialized in the Biomedical domain.

One of the most popular examples is the Biomedical NER (BioNER), whose focus is to extract relevant Biomedical entities, such as proteins, chemical compounds, or organism names from Biomedical documents.

Several BioNER corpora exist in the literature (see [92] for a recent comprehensive analysis). Most of them are publicly available on GitHub.[18] The characteristics of these corpora are briefly described in Table 3.

The BioNER tasks organized in the context of Biocreative[19] community effort are particularly relevant. Biocreative started in 2005 and the last edition was organized in 2017. Several corpora, such as BC5CDR or CHEMDNER, have been used in various organized challenges.

Additionally, there are a number of tasks derived from simple BioNER, such as (i) Concept Recognition, that assigns an identifier to retrieved entities, or (ii) Relation Extraction, which extracts relations between different entities, as described in Fig. 11. The information (and the supervision) for solving these tasks is usually included in BioNER corpora.

Biomedical text mining is not limited to BioNER. A further popular and relevant task is known as Biomedical Semantic Indexing (BioSI), and it consists in classifying Biomedical documents according to a hierarchy labels (e.g., MeSH).

Several BioSI shared-tasks have been organized by the BioASQ community[20] since 2012. The community provides large-scale corpora consisting of tens of millions of documents manually annotated by experts, representing the most important resources in BioSI [93].

Another relevant application is Biomedical QA, whose goal is to answer Biomedical-related questions. Curated resources manually annotated by experts are, for instance, PubMedQA [94] which contains a mixture of labelled, unlabelled, and artificially generated examples, and BioASQ [95]. However, these datasets are extremely small due to the cost of the annotation process.

Other works tried to build larger Biomedical QA corpora, e.g. [96,97]. However, questions of these corpora are mostly factoid, and answers can be extracted in the contexts without much reasoning. See [98] for a recent survey on Biomedical QA resources.

4.3.5. GLUE and other benchmarks

The General Language Understanding Evaluation (GLUE) benchmark [99] is a collection of tasks for evaluating models' performance on a set of various Natural Language Understanding (NLU) tasks in English.

GLUE consists of 9 different tasks, each one with different characteristics and peculiarities. In practice GLUE includes both single-sentence (linguistic acceptability, sentiment analysis) and sentence-pair tasks (similarity and paraphrase, linguistic inference). Moreover, the size of the corresponding datasets varies widely, ranging from a few hundred to a few hundred thousand examples. Additionally, the content of the datasets is extracted from several domains, including news, social media, books, and Wikipedia. Finally, the evaluation metric varies with the task and the dataset characteristics. Table 4 shows a synthetic representation of the benchmark features.

Given the wide variety of the benchmark, GLUE is regarded as a tool to evaluate models' ability to learn general linguistic knowledge and has become increasingly relevant in the Natural Language Processing community for general-purpose models evaluation. A public leaderboard is also available.[21]

However, the performance of state-of-the-art models has recently come close to the level of non-expert humans, suggesting limited headroom of GLUE for further research. In order to overcome this limitation, SuperGLUE [100] has recently been proposed. Along the lines of GLUE, SuperGLUE consists of several challenging NLU tasks, summarized in Table 5.

Inspired by the success of the General Language Understanding Evaluation benchmark, the Biomedical Language Understanding Evaluation (BLUE) benchmark [101] has been introduced to facilitate research in the biomedicine domain. The benchmark consists of five tasks with ten datasets that cover both biomedical and clinical texts with different dataset sizes and difficulties. BLUE relies on existing datasets that have been widely used by the BioNLP community as shared tasks. The five tasks are: sentence similarity, named entity recognition, relation extraction, document multilabel classification, and inference. A summary of these tasks is shown in Table 6.

[14] http://opus.nlpl.eu/ParaCrawl.php.
[15] http://opus.nlpl.eu/OpenSubtitles-v2018.php.
[16] http://opus.nlpl.eu/opus-100.php.
[17] http://www.ted.com/.
[18] https://github.com/cambridgeltl/MTL-Bioinformatics-2016.
[19] https://biocreative.bioinformatics.udel.edu/tasks/biocreative-vi/.
[20] http://bioasq.org.
[21] https://gluebenchmark.com.

Table 3
BNER corpora description.

Corpus	Entity counts
BC2GM	Gene/protein: 24,583
BC4CHEMD	Chemical: 84,310
BC5CDR	Chemical: 15,935; Disease: 12,852
BioNLP09	Gene/protein: 14,963
BioNLP11EPI	Gene/protein: 15,811
BioNLP11ID	Gene/protein: 6551 Organism: 3471 Chemical: 873 Regulon-operon: 87
CRAFT	Sequences: 18,974; Gene/protein:16,064; Chemical: 6053...
Linnaeus	Species: 4263
NCBI-disease	Disease: 6881

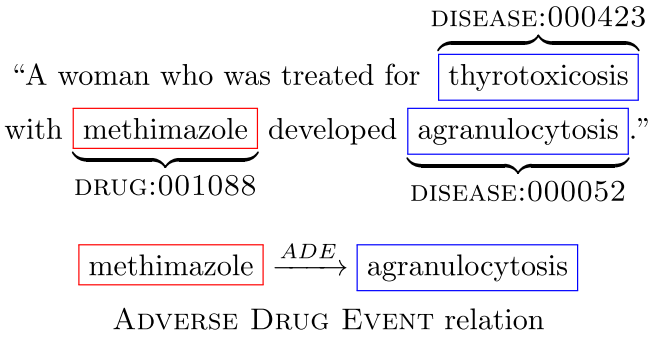


Fig. 11. Example of BioNER, Concept Recognition, and Relation Extraction. The BioNER module identifies *thyrotoxicosis*, *methimazole*, and *agranulocytosis* as named entities, the Concept Recognition module associates those entities to known identifiers, and the Relation Extraction module shows the relation between entities.

4.3.6. Other resources

We summarize here other resources and tasks which are not described in the previous sections.

NER was one of the first NLP tasks that started adopting quantitative evaluation in the context of competitions organized in the Message Understanding Conferences (MUCs) [66] mainly during the 1990's. Relevant entities in open-domain texts are usually persons, locations, or organizations. From the very beginning, languages other than English were taken into account: MUC (English, Japanese and Chinese), CoNLL 2002 (Spanish and Dutch; [102]), CoNLL 2003 (English and German; [103]), Automatic content extraction (ACE[22]; English, Arabic and Chinese) [104]. However, notwithstanding the interest NER has aroused in the past years, nowadays this task attracts researchers mainly in the Biomedical domain, which is considered much more challenging.

The shared tasks on Word Sense Disambiguation (WSD) started in the late 1990's with the pilot evaluation exercise at SensEval [105] in 1998. At SensEval-1 the languages considered were English, French, and Italian. In the following editions of SensEval/SemEval, WSD tasks with different characteristics and dealing with different languages were present till SemEval 2015 Task 13[23] (Multilingual All-Words Sense Disambiguation and Entity Linking).

Part-of-Speech (PoS) tagging is the process of marking up a word in a text as corresponding to a particular part of speech, e.g. verbs, adjectives, articles, and so on. The first major corpus of English for PoS tagging was the Brown Corpus, developed in the mid-1960s. This corpus has been used for several studies concerning word-frequency and POS, and it inspired the development of similar annotated corpora in many other languages. Another standard dataset for POS tagging is the Wall Street Journal (WSJ) portion of the Penn Treebank [106], containing 45 different POS tags and 5 millions of annotated tokens.

As is the case of other tasks, PoS is suffering a decline in interest after the advent of state-of-the-art deep learning models. Nowadays, the research is moving towards critical scenarios, such as historical corpora [107], languages with poor annotated data, and cross-languages [108].

5. Current issues and future directions

Word and sentence/document embeddings are constantly evolving, and new representations are continuously proposed. However, despite the capabilities of this new generation of models, there are still problems in NLP that need to be solved. E.g., popular Transformers are not fully able to encode whole documents as their complexity is quadratic to the sequence length and their input sequences cannot exceed 512 tokens, which typically corresponds to a single paragraph. However, a few variations of the classical Transformer exist to handle long documents, as is the case of Transformer-XL [45] and Longformer [109], which apply a relaxation of the classical self-attention with various strategies.

Moreover, one of the main worrying aspects of these models is their computational cost. Pre-trained transformers usually consist of 110–340 million of learnable parameters, and they require specialized and expensive hardware. As we already mentioned, this last topic is becoming popular in the NLP community and it is extremely important as the carbon footprint of training such model is growing exponentially. Consequently, a considerable branch of research is currently exploring the development of efficient methods, including lighter Transformers [110,41,111] and distillation approaches [42,112].

The computational cost described above negatively affects the model selection, and the exhaustive evaluation of multiple hyper-parameters (learning rate, batch size, warm-up scheduler...) configurations can be hardly carried out. In order to alleviate this issue, the model selection is often simplified by using only default configurations or only a single run. The main consequence, is that the quality of scientific results may significantly decrease as they may drop reproducibility. The statistical significance, for example, is rarely taken into account. In order to better explore this aspect, we manually analyzed a set of 50 peer-reviewed scientific papers that use a pre-trained Transformer model. These papers have been randomly sampled from the ACL-Anthology, a popular repository for NLP papers. Results of our analysis indicates that only 26% of papers declared that multiple fine-tuning runs of the Transformer have been considered, but only 10% reported the standard deviation. Moreover, 80% of papers do not show a complete and clear model selection procedure, i.e. they did not report the optimal configuration or they simply exposed the used configuration, without mentioning the searching strategies nor the tested values. To make things worse, it is known that these architectures may be really sensitive to the selection of the hyper-parameters [113,114].

As a further limitation, Transformers and modern architectures are often used as blackbox tools, and their outputs are hardly interpretable. Interpretability is a key aspect of NLP applications in delicate domains like medicine for instance. Interpretability is becoming a newsworthy aspect in the literature, and it has been the main topic of several recent workshops.[24] The goal of the Black-

[22] https://web.archive.org/web/20060308054306/http://www.itl.nist.gov/iad/894.01/tests/ace/

[23] https://alt.qcri.org/semeval2015/task13/.

[24] E.g. BlackboxNLP, held at EMNLP 2018, ACL 2019 and EMNLP 2020.

Table 4
GLUE benchmark features.

Corpus	|Train|	Task	Metric	Domain
CoLA	8.5 K	acceptability	Matthews corr	misc.
SST-2	67 K	sentiment	accuracy	movie reviews
MRPC	3.7 K	paraphrase	accuracy/F_1	news
STS-B	7 K	sent. similarity	Pearson/Spearman corr	misc.
QQP	364 K	paraphrase	accuracy/F_1	social questions
MNLI	393 K	NLI	matched/mismatched acc	misc.
QNLI	105 K	QA/NLI	accuracy	Wikipedia
RTE	2.5 K	NLI	accuracy	news, Wikipedia
WNLI	634	coref./NLI	accuracy	fiction book

Table 5
SuperGLUE benchmark features.

Corpus	|Train|	Task	Metric	Domain
BoolQ	9427	yes/no QA	acc.	Google queries, Wikipedia
CB	250	NLI	acc./F_1	misc.
COPA	400	QA	acc.	blogs, photography encyclopedia
MultiRC	5100	QA	F1a/EM	misc.
ReCoRD	101 K	QA	F_1/EM	news (CNN, Daily Mail)
RTE	2500	NLI	acc.	news, Wikipedia
WiC	6000	WSD	acc.	WordNet, VerbNet, Wikitionary
WSC	554	coref.	acc.	fiction books

Table 6
BLUE benchmark features.

Corpus	|Train|	Task	Metric	Domain
MedSTS,sentence pairs	675	sent. similarity	Pearson	Clinical
BIOSSES,sentence pairs	64	sent. similarity	Pearson	Biomedical
BC5CDR-disease,mentions	4,182	NER	F_1	Biomedical
BC5CDR-chemical,mentions	5,203	NER	F_1	Biomedical
ShARe/CLEFE,mentions	4,628	NER	F_1	Clinical
DDI,relations	2,937	relation extraction	micro-F_1	Biomedical
ChemProt,relations	4,154	relation extraction	micro-F_1	Biomedical
i2b2 2010,relations	3,110	relation extraction	F_1	Clinical
HoC,documents	1,108	document classification	F_1	Biomedical
MedNLI,pairs	11,232	inference	accuracy	Clinical

boxNLP workshop series is to bring together people who are attempting to peek inside the neural network blackbox, taking inspiration from machine learning, psychology, linguistics, and neuroscience. Finally, further research directions in NLP include (i) cyber-security, such as fake news detection, (ii) industrial applications, such as virtual assistants (Alexa, Siri...), and (iii) text generation based for instance on recent variational autoencoders or Generative Adversarial Networks.

CRediT authorship contribution statement

Ivano Lauriola: Writing - original draft, Writing - review & editing. **Alberto Lavelli:** Supervision. **Fabio Aiolli:** Supervision.

Declaration of Competing Interest

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

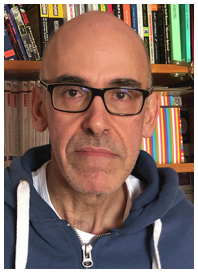
References

[1] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, CoRR (abs/1409.0473.).
[2] V. Yadav, S. Bethard, A survey on recent advances in named entity recognition from deep learning models, in: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 2145–2158, URL: https://www.aclweb.org/anthology/C18-1182.
[3] P. Rajpurkar, R. Jia, P. Liang, Know what you don't know: Unanswerable questions for SQuAD, in: Proc. of ACL, 2018.
[4] P. Capuozzo, I. Lauriola, C. Strapparava, F. Aiolli, G. Sartori, Decop: A multilingual and multi-domain corpus for detecting deception in typed text, in: Proceedings of The 12th Language Resources and Evaluation Conference, 2020, pp. 1423–1430.
[5] K. Jing, J. Xu, A survey on neural network language models (2019). arXiv:1906.03591..
[6] S. Ruder, I. Vulić, A. Søgaard, A survey of cross-lingual word embedding models, J. Artif. Int. Res. 65 (1) (2019) 569–630.
[7] A. Torfi, R.A. Shirvani, Y. Keneshloo, N. Tavaf, E.A. Fox, Natural language processing advancements by deep learning: A survey (2021). arXiv:2003.01200..
[8] D.W. Otter, J.R. Medina, J.K. Kalita, A survey of the usages of deep learning for natural language processing, IEEE Transactions on Neural Networks and Learning Systems 32 (2) (2021) 604–624.
[9] K. Babic, S. Martincic-Ipšic, A. Meštrovic, Survey of neural text representation models, Information 11 (11) (2020) 511.
[10] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: NIPS, 2013.
[11] S.K. Sienčnik, Adapting word2vec to named entity recognition, in: Proc. of the 20th Nordic Conference of Computational Linguistics, 2015.
[12] B. Xue, C. Fu, Z. Shaobin, A study on sentiment computing and classification of sina weibo with word2vec, in: IEEE International Congress on Big Data, IEEE, 2014..
[13] H. Caselles-Dupré, F. Lesaint, J. Royo-Letelier, Word2vec applied to recommendation: Hyperparameters matter, in: Proc. of the 12th ACM Conference on Recommender Systems, 2018.
[14] T. Dao, S. Keller, A. Bejnood, Alternate equivalent substitutes: Recognition of synonyms using word vectors, Tech. rep., Stanford University, 2013.
[15] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proc. of EMNLP, 2014.

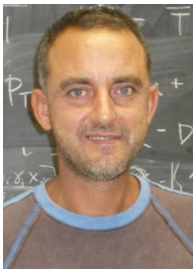
[16] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, T. Mikolov, Learning word vectors for 157 languages, in: Proc. of LREC, 2018..
[17] M. Basaldella, L. Furrer, C. Tasso, F. Rinaldi, Entity recognition in the biomedical domain using a hybrid approach, Journal of Biomedical Semantics 8 (1) (2017) 51.
[18] F. Almeida, G. Xexéo, Word embeddings: A survey, arXiv preprint arXiv:1901.09069..
[19] Z.S. Harris, Distributional structure, Word 10 (2–3) (1954) 146–162.
[20] G.V. Cormack, J.M. Gómez Hidalgo, E.P. Sánz, Spam filtering for short messages, in: Proc. of the Sixteenth ACM Conference on Information and Knowledge Management, 2007.
[21] E. Gabrilovich, S. Markovitch, Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4.5, in: Proc. of ICML, 2004.
[22] R. Bekkerman, J. Allan, Using bigrams in text categorization, Tech. rep., Technical Report IR-408, Center of Intelligent Information Retrieval (2004)..
[23] R. Liu, D. Wang, C. Xing, Document classification based on word vectors, in: Proc. of ISCSLP, 2014.
[24] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: Proc. of ICML, 2014..
[25] R. Kiros, Y. Zhu, R.R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, S. Fidler, Skip-thought vectors, in: Proc. of NIPS, 2015..
[26] F. Hill, K. Cho, A. Korhonen, Learning distributed representations of sentences from unlabelled data, in: Proc. of NAACL-HLT, 2016.
[27] Z. Gan, Y. Pu, R. Henao, C. Li, X. He, L. Carin, Learning generic sentence representations using convolutional neural networks, in: Proc. of EMNLP, 2016.
[28] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proc. of EMNLP, 2013.
[29] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, in: Proc. of ACL, 2014.
[30] Y. Kim, Convolutional neural networks for sentence classification, in: Proc. of EMNLP, 2014.
[31] A. Severyn, A. Moschitti, Learning to rank short text pairs with convolutional deep neural networks, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015, pp. 373–382.
[32] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Proc. of NIPS, 2014.
[33] K. Cho, B. van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder–decoder approaches, in: Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, 2014.
[34] P. Koehn, F.J. Och, D. Marcu, Statistical phrase-based translation, in: Proc. of HLT-NAACL, 2003.
[35] O. Melamud, J. Goldberger, I. Dagan, context2vec: Learning generic context embedding with bidirectional LSTM, in: Proc. of the 20th SIGNLL Conference on Computational Natural Language Learning, 2016.
[36] R. Józefowicz, O. Vinyals, M. Schuster, N. Shazeer, Y. Wu, Exploring the limits of language modeling, ArXiv abs/1602.02410..
[37] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proc. of NAACL-HLT, 2018.
[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Proc. of NIPS, 2017..
[39] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186..
[40] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, ArXiv abs/1907.11692..
[41] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A lite BERT for self-supervised learning of language representations, arXiv preprint arXiv:1909.11942..
[42] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108..
[43] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training, URL: https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/ language understanding paper. pdf..
[44] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, OpenAI Blog 1 (8) (2019) 9.
[45] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, R. Salakhutdinov, Transformer-XL: Attentive language models beyond a fixed-length context, in: Proc. of ACL, 2019.
[46] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R.R. Salakhutdinov, Q.V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, in: Proc. of NIPS, 2019.
[47] T. Shao, Y. Guo, H. Chen, Z. Hao, Transformer-based neural network for answer selection in question answering..
[48] S. Garg, T. Vu, A. Moschitti, Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 7780–7788..
[49] S. Kumar, K. Mehta, N. Rasiwasia, et al., Improving answer selection and answer triggering using hard negatives, in: EMNLP-IJCNLP, 2019.
[50] C. Sun, X. Qiu, Y. Xu, X. Huang, How to fine-tune BERT for text classification?, in: China National Conference on Chinese Computational Linguistics, Springer, 2019, pp 194–206.
[51] M. Hoang, O.A. Bihorac, J. Rouces, Aspect-based sentiment analysis using BERT, in: Proc. of NoDaLiDa, 2019.
[52] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C.H. So, J. Kang, BioBERT: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (4) (2020) 1234–1240.
[53] X. Yu, W. Hu, S. Lu, X. Sun, Z. Yuan, BioBERT based named entity recognition in electronic medical record, in: 2019 10th International Conference on Information Technology in Medicine and Education (ITME), 2019, pp. 49–52.
[54] T. Pires, E. Schlinger, D. Garrette, How multilingual is multilingual BERT?, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp 4996–5001.
[55] Z. Wang, S. Mayhew, D. Roth, et al., Cross-lingual ability of multilingual bert: An empirical study, ICLR..
[56] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6–12, 2020, virtual, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html..
[57] S. Bird, E. Loper, E. Klein, Natural language processing with python o'reilly media inc..
[58] R. Řeh∘uřek, P. Sojka, Software Framework for Topic Modelling with Large Corpora, in: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, ELRA, Valletta, Malta, 2010, pp. 45–50..
[59] M. Honnibal, I. Montani, S. Van Landeghem, A. Boyd, spaCy: Industrial-strength Natural Language Processing in Python (2020)..
[60] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Funtowicz, J. Davison, S. Shleifer, 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, pp. 38–45.
[61] C.D. Manning, M. Surdeanu, J. Bauer, J.R. Finkel, S. Bethard, D. McClosky, The stanford corenlp natural language processing toolkit, in: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014, pp. 55–60.
[62] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C.D. Manning, Stanza: A Python natural language processing toolkit for many human languages, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2020, URL: https://nlp.stanford.edu/pubs/qi2020stanza.pdf.
[63] E. Strubell, A. Ganesh, A. McCallum, Energy and policy considerations for deep learning in NLP, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 3645–3650.
[64] L.F.W. Anthony, B. Kanding, R. Selvan, Carbontracker: Tracking and predicting the carbon footprint of training deep learning models, arXiv preprint arXiv:2007.03051..
[65] R. Schwartz, J. Dodge, N.A. Smith, O. Etzioni, Green ai, arXiv preprint arXiv:1907.10597..
[66] L. Hirschman, The evolution of evaluation: Lessons from the message understanding conferences, Computer Speech and Language 12 (1998) 281–305.
[67] D. Mollá, J.L. Vicedo, Question answering in restricted domains: An overview, Computational Linguistics 33 (1) (2007) 41–61.
[68] W.-T. Yih, M.-W. Chang, C. Meek, A. Pastusiak, Question answering using enhanced lexical semantic models, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 1744–1753. URL: https://www.aclweb.org/anthology/P13-1171..
[69] W. Bian, S. Li, Z. Yang, G. Chen, Z. Lin, A compare-aggregate model with dynamic-clip attention for answer selection, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 1987–1990.
[70] S. Yoon, F. Dernoncourt, D.S. Kim, T. Bui, K. Jung, A compare-aggregate model with latent clustering for answer selection, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 2093–2096.
[71] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, SQuAD: 100,000+ questions for machine comprehension of text, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing doi:10.18653/v1/d16-1264. URL: https://doi.org/10.18653/v1/D16-1264..
[72] P. Rajpurkar, R. Jia, P. Liang, 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), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 784–789.

[73] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, et al., Natural questions: a benchmark for question answering research, Transactions of the Association for Computational Linguistics 7 (2019) 453–466.
[74] Y. Yang, W.-T. Yih, C. Meek, A challenge dataset for open-domain question answering, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 2013–2018.
[75] E. Choi, H. He, M. Iyyer, M. Yatskar, W.-T. Yih, Y. Choi, P. Liang, L. Zettlemoyer, QuAC: Question answering in context, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 2174–2184.
[76] S. Reddy, D. Chen, C.D. Manning, CoQA: A conversational question answering challenge, Transactions of the Association for Computational Linguistics 7 (2019) 249–266.
[77] H. Zamani, S. Dumais, N. Craswell, P. Bennett, G. Lueck, Generating clarifying questions for information retrieval, in: Proceedings of The Web Conference 2020, WWW '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 418–428..
[78] H. Zamani, G. Lueck, E. Chen, R. Quispe, F. Luu, N. Craswell, Mimics: A large-scale data collection for search clarification, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 3189–3196.
[79] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, L. Deng, Ms marco: A human-generated machine reading comprehension dataset, in: Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches, 2016.
[80] A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, C. Potts, Learning word vectors for sentiment analysis, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 142–150. URL: http://www.aclweb.org/anthology/P11-1015..
[81] A. Go, R. Bhayani, L. Huang, Twitter sentiment classification using distant supervision, CS224N project report, Stanford 1 (12) (2009) 2009.
[82] B. Keith, E. Fuentes, C. Meneses, A hybrid approach for sentiment analysis applied to paper reviews..
[83] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J.N. Chang, S. Lee, S.S. Narayanan, Iemocap: Interactive emotional dyadic motion capture database, Language Resources and Evaluation 42 (4) (2008) 335.
[84] A. Zadeh, R. Zellers, E. Pincus, L.-P. Morency, Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos, arXiv preprint arXiv:1606.06259..
[85] A. Bagher Zadeh, P.P. Liang, S. Poria, E. Cambria, L.-P. Morency, Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 2236–2246.
[86] P. Koehn, Europarl: A parallel corpus for statistical machine translation, in: Proc. of MT Summit 2005, 2005.
[87] J. Tiedemann, Parallel data, tools and interfaces in OPUS, in: Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC, 2012.
[88] P. Lison, J. Tiedemann, Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles, in: N.C.C. Chair), K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), European Language Resources Association (ELRA), Paris, France, 2016..
[89] M. Cettolo, C. Girardi, M. Federico, WIT3: Web inventory of transcribed and translated talks, in: Proceedings of the 16th Annual conference of the European Association for Machine Translation, European Association for Machine Translation, Trento, Italy, 2012, pp. 261–268. URL: https://www.aclweb.org/anthology/2012.eamt-1.60..
[90] R. Cattoni, M.A.D. Gangi, L. Bentivogli, M. Negri, M. Turchi, MuST-C: A multilingual corpus for end-to-end speech translation, Computer Speech and Language..
[91] J. Iranzo-Sánchez, J.A. Silvestre-Cerdà, J. Jorge, N. Roselló, A. Giménez, A. Sanchis, J. Civera, A. Juan, Europarl-st: A multilingual corpus for speech translation of parliamentary debates, in: ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 8229–8233.
[92] G. Crichton, S. Pyysalo, B. Chiu, A. Korhonen, A neural network multi-task learning approach to biomedical named entity recognition, BMC Bioinformatics 18 (1) (2017) 368.
[93] A. Nentidis, A. Krithara, K. Bougiatiotis, M. Krallinger, C. Rodriguez-Penagos, M. Villegas, G. Paliouras, Overview of bioasq 2020: The eighth bioasq challenge on large-scale biomedical semantic indexing and question answering, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2020, pp. 194–214.
[94] Q. Jin, B. Dhingra, Z. Liu, W.W. Cohen, X. Lu, Pubmedqa: a dataset for biomedical research question answering, arXiv preprint arXiv:1909.06146..
[95] A. Nentidis, A. Krithara, K. Bougiatiotis, G. Paliouras, Overview of BioASQ 8a and 8b: Results of the eighth edition of the BioASQ tasks a and b, in: Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, 2020.
[96] A. Pampari, P. Raghavan, J. Liang, J. Peng, emrQA: A large corpus for question answering on electronic medical records, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 2357–2368, URL: https://www.aclweb.org/anthology/D18-1258.
[97] D. Pappas, I. Androutsopoulos, H. Papageorgiou, BioRead: A new dataset for biomedical reading comprehension, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Miyazaki, Japan, 2018. URL: https://www.aclweb.org/anthology/L18-1439..
[98] M. Wasim, W. Mahmood, U.G. Khan, A survey of datasets for biomedical question answering systems, International Journal of Advanced Computer Science and Applications 8 (7) (2017) 484–488.
[99] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman, 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, pp. 353–355.
[100] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman, Superglue: A stickier benchmark for general-purpose language understanding systems, in: Advances in Neural Information Processing Systems, 2019, pp. 3266–3280..
[101] Y. Peng, S. Yan, Z. Lu, Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets, in: Proceedings of the 18th BioNLP Workshop and Shared Task, Association for Computational Linguistics, Florence, Italy, 2019, pp. 58–65.
[102] E.F. Tjong Kim Sang, Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition, in: COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002), 2002. URL: https://www.aclweb.org/anthology/W02-2024.
[103] E.F. Tjong Kim Sang, F. De Meulder, 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, pp. 142–147. URL: https://www.aclweb.org/anthology/W03-0419..
[104] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, R. Weischedel, The automatic content extraction (ACE) program – tasks, data, and evaluation, in: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04), European Language Resources Association (ELRA), Lisbon, Portugal, 2004, URL: http://www.lrec-conf.org/proceedings/lrec2004/pdf/5.pdf.
[105] A. Kilgariff, Senseval: An exercise in evaluating word sense disambiguation programs, Proc. of LREC (1998).
[106] M. Marcus, B. Santorini, M.A. Marcinkiewicz, Building a large annotated corpus of english: The penn treebank..
[107] S. Heid, M. Wever, E. Hüllermeier, Reliable part-of-speech tagging of historical corpora through set-valued prediction, arXiv preprint arXiv:2008.01377..
[108] J.-K. Kim, Y.-B. Kim, R. Sarikaya, E. Fosler-Lussier, Cross-lingual transfer learning for pos tagging without cross-lingual resources, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 2832–2838.
[109] I. Beltagy, M.E. Peters, A. Cohan, Longformer: The long-document transformer, arXiv preprint arXiv:2004.05150..
[110] N. Kitaev, Ł. Kaiser, A. Levskaya, Reformer: The efficient transformer, arXiv preprint arXiv:2001.04451..
[111] A. de Wynter, D.J. Perry, Optimal subarchitecture extraction for bert, arXiv preprint arXiv:2010.10499..
[112] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531..
[113] M. Leszczynski, A. May, J. Zhang, S. Wu, C.R. Aberger, C. Ré, Understanding the downstream instability of word embeddings, arXiv preprint arXiv:2003.04983..
[114] X. Zhou, Y. Nie, H. Tan, M. Bansal, The curse of performance instability in analysis datasets: Consequences, source, and suggestions, arXiv preprint arXiv:2004.13606..

Ivano Lauriola, after receiving his Master's Degree in Computer Science at the University of Padova, pursued a Ph.D. at the Department of Mathematics, University of Padova. His research is mainly focused on Kernel Learning methods, including Multiple Kernel Learning, Deep Kernels, and applications to Natural Language Processing tasks. Currently he is an applied scientist at Amazon Alexa AI.



Alberto Lavelli received a Master's Degree in Computer Science from the University of Milano. Currently he is a Senior Researcher at Fondazione Bruno Kessler in Trento (Italy). His main research interests concern the application of machine learning techniques to Information Extraction from text, in particular in the biomedical domain.



Fabio Aiolli received a Master's Degree and a PhD in Computer Science both from the University of Pisa. He is currently Associate Professor at the University of Padova. His research activity is mainly in the area of Machine Learning and Information Retrieval.