ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Language processing in the era of deep learning

Ivano Lauriola[1,2]*, Alberto Lavelli[2], and Fabio Aiolli[1]

1- University of Padova - Dept of Mathematics
Via Trieste, 63, 35121 Padova - Italy

2- Fondazione Bruno Kessler
Via Sommarive, 18, 38123 Trento - Italy

Abstract. Natural Language Processing is a branch of artificial intelligence brimful of intricate, sophisticated, and challenging tasks, such as machine translation, question answering, summarization, and so on. 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. 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 this contribution, we provide a high-level overview of recent advances in NLP, the role of Machine Learning, and current research directions.

1 Introduction

The field of Natural Language Processing (NLP) involves the design and implementation of computational models and processes to solve practical problems in understanding human languages. On the one hand, work in NLP addresses fundamental problems such as language modeling, morphological analysis, syntactic processing, or parsing, and semantic analysis. On the other hand, 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. Currently, NLP is primarily a data-driven field using statistical and probabilistic computations along with Machine Learning (ML). In the past, machine learning approaches such as naive Bayes, k-nearest neighbors, Hidden Markov Models (HMMs), Conditional Random Fields (CRFs), decision trees, random forests, and Support Vector Machines (SVMs) were widely used. However, during the past several years, there has been a wholesale transformation, and these approaches have been entirely replaced, or at least enhanced, by neural models, discussed later.

The deep learning revolution has influenced and changed many fields of Artificial Intelligence (e.g., ML and computer vision) and has also affected all areas related to human language technologies. Initial results have been obtained with the adoption of deep neural networks in speech recognition, with a significant boost of performance in automatic speech recognition systems [1]. In Machine

*The work of I. Lauriola is partially supported by grant CR30I1_162758 of the Swiss National Science Foundation.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Translation, starting from 2013, the phrase-based statistical approaches that were at the state of the art have been gradually substituted with neural machine translation, based on deep learning architectures, which obtained better performance [2]. The main reason for this increase of performance is that, as more training data are available both for speech recognition and machine translation, large neural networks have demonstrated to be superior to traditional ML algorithms, such as SVM. 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. It is only in the last few years that deep learning approaches have obtained very high performance across many different NLP tasks. Because of the fact that these models can often be trained with a single end-to-end model and do not require traditional, task-specific feature engineering, they not only tend to perform better than traditional ML, but they do require less human effort, making their adoption convenient.

Let us consider what has happened in the information extraction (IE) field. The earliest work on IE addressed the template-filling task in the context of the U.S. government-sponsored MUC conferences, where the standard evaluation techniques were defined. The standard approaches were based on manually written rules. Due to the difficulty of porting systems from one domain to another, attention shifted to ML approaches. Early supervised learning approaches to IE focused on automating the knowledge acquisition process, mainly for finite-state rule-based systems. Their success, and the earlier success of HMM-based speech recognition, led to the use of sequence labeling (HMMs, CRFs), and wide exploration of features. Neural approaches to Named Entity Recognition (NER) mainly follow from the pioneering results of [3], who applied a CRF on top of a convolutional net. BiLSTMs with word and character-based embeddings as input followed shortly and became a standard neural algorithm for NER [4, 5, 6]. Progress in this area continues to be stimulated by formal evaluations with shared benchmark datasets, including the Automatic Content Extraction (ACE) evaluations of 2000-2007 on named entity recognition, relation extraction, and temporal expressions, the KBP (Knowledge Base Population) evaluations [7, 8] of relation extraction tasks like slot filling (extracting attributes like age, birthplace, and spouse for a given entity) and a series of SemEval workshops [9].

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 3 classes[1]:

Sequence classification: Let $\mathcal{S}$ be a set of sequences, where each sequence $s \in \mathcal{S}$ is a series of tokens $s = \langle w_1 \dots w_{|s|} \rangle$ and let $\mathcal{C} = \{c_1, c_2 \dots\}$ be a set of possible classes. Similarly to common classification problems in ML, the aim of sequence classification is to find a function $f : \mathcal{S} \to \mathcal{C}$ able to

[1] This classification is not exhaustive but covers most of the popular and relevant tasks.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

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.

Word labeling: In word labeling applications a label is output from each token $w_i \in s$. Examples of word labeling tasks are (i) 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, adjective...).

sequence2sequence In seq2seq problems the input sequence is used to generate an output sequence. Differently from word labeling applications, the input and output sequences are not directly aligned, and the model needs to *generate* a new sentence. The canonical example is machine translation.

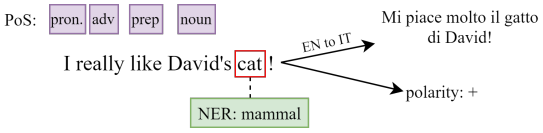

Fig. 1: Examples of NLP tasks applied to the same input sentence, including PoS tagging, NER (focusing on species), sentiment analysis, and translation.

3 Recent advances in NLP

One of the main problems in the last decade 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 $\mathcal{D}$ as one-hot encoding $\boldsymbol{h}_{w_i} \in \{0,1\}^{|\mathcal{D}|}$. 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 strongly limited the capability of NLP systems, unable for instance to catch the similarities between *apple*, *kiwi*, *table*, *peach* words and to find the unrelated one.

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, for which words with similar meaning have a similar representation. 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 $\boldsymbol{h}_{w_i} \xrightarrow{enc} \boldsymbol{v}_{w_i} \xrightarrow{dec} \boldsymbol{h}_{w_j}$, 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

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

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 $\boldsymbol{v}_{king} - \boldsymbol{v}_{queen} = \boldsymbol{v}_{man} - \boldsymbol{v}_{woman}$.

Due to its effectiveness on several tasks, such as NER [11, 12], sentiment analysis [13], recommendation [14], and synonym recognition [15], Word2vec received considerable attention in the literature, and several improved solutions have subsequently been proposed. Some relevant examples are (i) Global-Vector (GloVe) [16], that exploits statistical information computed on the whole corpus, and (ii) fastText [17], 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 [18, 12], where for instance affixes of biomedical terms have a specific structure.

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 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 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 has been used. 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

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

them are based on recursive [28], recurrent, or convolutional neural networks [29, 30]. 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.

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 [31, 32]. Unlike the previous phrase-based translation system [33] 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. 2 left), named *context* vector, with a consequent drop in performance. To this end, attention mechanisms [2] have been introduced, where the context vector used to produce each output state is defined as a linear combination of all internal encoding contexts (Fig. 2 right). The model showed remarkable results when dealing with long sentences.

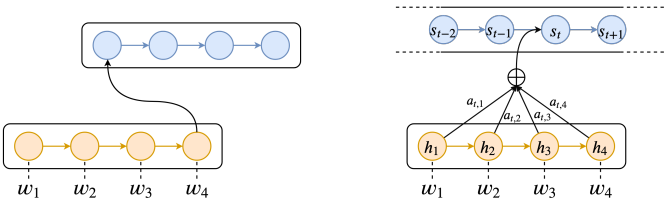

Fig. 2: Left: classical sequence2sequence architecture based on recurrent neural networks. The encoder (orange) produces a context vector to fed the decoder (blue). Right - the attention mechanism allows to produce an output state by means of a combination of intermediate context vectors.

Inspired by the recent success of bidirectional RNN [34, 35], ELMo [36] (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 pre-trained on unlabeled data. Most important, authors showed mechanisms to use internal representations in downstream tasks by fine-tuning the network, improving results on several benchmarks.

However, the last real boost in NLP after the advent of word vectors and unsupervised pre-training is the Transformer model [37]. 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

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

the most popular pre-trained Transformer models is BERT [38] (Bidirectional Encoder Representations from Transformers). BERT is designed to pre-train deep bidirectional representations from unlabeled text 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, the network masks a small amount of words of the input sequence and it tries to predict them, 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 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 [39], ALBERT [40], and DistilBERT [41]. 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 humans, and it does not improve the performance of the network. Other relevant methods based on the same concepts are GPT [42] (Generative Pre-Training), GPT-2 [43], Transformer-XL [44], and its extension XLNet [45].

Nowadays, these methods are continuously achieving excellent performance on a plethora of NLP tasks, such as question answering [46, 38], text classification [47], and sentiment analysis [48]. Surprisingly, these networks started to overcome human performance on several tasks that were considered unsolvable by AI, such as Question Answering [49] and verbal lie detection [50].

4 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 able to encode whole documents as their input sequences cannot exceed 512 tokens, that typically corresponds to a single paragraph. 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. Furthermore, the model selection covers several sensitive hyper-parameters, such as the learning rate, batch size, and warm-up scheduler, making the approach hardly practicable on large-scale datasets, as is the case of question answering and text generation. To this end, a considerable branch of research is currently exploring the development of efficient methods, including lighter Transformers

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

[51, 40] and distillation approaches [41, 52].

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[2]. 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.

5 Contributions to ESANN 2020

The contributions in this special session cover several tasks and applications in NLP. [53], Papers [54] and [55] investigated applications and extensions of word vectors. LSTM language models are evaluated in [56] and [57] from different points of view. Finally, authors in [58] analyzed adversarial attacks in speech recognition. The s.s. received contribution from both, companies and academies, showing that NLP is becoming an hot-topic also for industrial purposes.

In more detail, the work in [56] investigated how well LSTM language models can leverage on long-term contexts. This is made by extending the language model inspired by the Word2vec CBOW model. Language modeling and speech recognition experiments on English and Dutch data sets have been carried out, showing that LSTM LMs are inherently capable of learning basic semantic information of a limited history, i.e. the context. Authors in [53] explored word meta-embeddings defined as combinations of two (and virtually more) pre-trained word vectors from GloVe and fastText. Interestingly, the proposed approach does not require the presence of the target word in both input embeddings. Authors also proposed an autoencoder-like architecture to learn the meta embedding, showing remarkable empirical results. A framework for controlling length in sentence generation has been proposed in [57]. The framework is based on a two-stage training. In the first stage a summarizer is trained without any explicit control to the sentence length. The second stage fine-tunes the summarized sentences by training a stylizer that adjusts the length. Experiments show that the proposed approach achieves comparable results with respect to the state of the art. Authors in [54] focused on NER and relation extraction, i.e. the task of finding relations between entities, in the biomedical domain. To this end, different approaches to learn entity and entity-pair embeddings have been explored, showing new state of the art results on the CHEMPROT corpus. Word embeddings have been analyzed also in [55]. Here, different word vectors based on character-level representations have been compared, including kernelized approaches. Furthermore, an extremely efficient ensemble based on Extreme Learning Machines and spectrum kernels has been proposed. Finally, authors in [58] described a methodology to detect adversarial attacks and to restore the original label for an attacked input in the context of speech recognition.

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

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

The proposed method is inspired by a similar work by the same authors in the context of image classification.

References

[1] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[3] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[4] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. arXiv.1508.01991 [cs.CL], 2015.

[5] Xuezhe Ma and Eduard Hovy. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proc. of ACL*, 2016.

[6] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proc. of NAACL-HLT*, 2016.

[7] Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. Overview of the TAC2010 Knowledge Base Population track. In *Proc. of the Third Text Analysis Conference (TAC)*, 2010.

[8] Mihai Surdeanu. Overview of the TAC2013 Knowledge Base Population evaluation: English slot filling and temporal slot filling. In *Proc. of the TAC-KBP 2013 Workshop*, 2013.

[9] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, and Diarmuid Ó Séaghdha et al. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proc. of NAACL Workshop on Semantic Evaluations*, 2009.

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[11] Scharolta Katharina Sienčnik. Adapting word2vec to named entity recognition. In *Proc. of the 20th nordic conference of computational linguistics*, 2015.

[12] Ivano Lauriola, Riccardo Sella, Fabio Aiolli, Alberto Lavelli, and Fabio Rinaldi. Learning representations for biomedical named entity recognition. In *2nd workshop on Natural Language for Artificial Intelligence (NL4AI)*, 2018.

[13] Bai Xue, Chen Fu, and Zhan Shaobin. A study on sentiment computing and classification of sina weibo with word2vec. In *IEEE International Congress on Big Data*. IEEE, 2014.

[14] Hugo Caselles-Dupré, Florian Lesaint, and Jimena Royo-Letelier. Word2vec applied to recommendation: Hyperparameters matter. In *Proc. of the 12th ACM Conference on Recommender Systems*, 2018.

[15] Tri Dao, Sam Keller, and Alborz Bejnood. *Alternate equivalent substitutes: Recognition of synonyms using word vectors*. PhD thesis, US: Stanford University, 2013.

[16] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proc. of EMNLP*, 2014.

[17] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proc. of LREC*, 2018.

[18] Marco Basaldella, Lenz Furrer, Carlo Tasso, and Fabio Rinaldi. Entity recognition in the biomedical domain using a hybrid approach. *Journal of biomedical semantics*, 8(1):51, 2017.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

[19] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[20] Gordon V Cormack, José María Gómez Hidalgo, and Enrique Puertas Sánz. Spam filtering for short messages. In *Proc. of the sixteenth ACM conference on Conference on information and knowledge management*, 2007.

[21] Evgeniy Gabrilovich and Shaul Markovitch. Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4.5. In *Proc. of ICML*, 2004.

[22] Ron Bekkerman and James Allan. Using bigrams in text categorization. Technical report, Technical Report IR-408, Center of Intelligent Information Retrieval, 2004.

[23] Rong Liu, Dong Wang, and Chao Xing. Document classification based on word vectors. In *Proc. of ISCSLP*. 2014.

[24] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proc. of ICML*, 2014.

[25] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Proc. of NIPS*, 2015.

[26] Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. In *Proc. of NAACL-HLT*, 2016.

[27] Zhe Gan, Yunchen Pu, Ricardo Henao, Chunyuan Li, Xiaodong He, and Lawrence Carin. Learning generic sentence representations using convolutional neural networks. In *Proc. of EMNLP*, 2016.

[28] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of EMNLP*, 2013.

[29] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proc. of ACL*, 2014.

[30] Yoon Kim. Convolutional neural networks for sentence classification. In *Proc. of EMNLP*, 2014.

[31] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Proc. of NIPS*, 2014.

[32] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014.

[33] Philipp Koehn, Franz J. Och, and Daniel Marcu. Statistical phrase-based translation. In *Proc. of HLT-NAACL*, 2003.

[34] Oren Melamud, Jacob Goldberger, and Ido Dagan. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proc. of the 20th SIGNLL Conference on Computational Natural Language Learning*, August 2016.

[35] Rafal Józefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *ArXiv*, abs/1602.02410, 2016.

[36] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL-HLT*, 2018.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. of NIPS*, 2017.

[38] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

[39] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv*, abs/1907.11692, 2019.

[40] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[41] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[42] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[43] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[44] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proc. of ACL*, 2019.

[45] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Proc. of NIPS*, 2019.

[46] Siddhant Garg, Thuy Vu, and Alessandro Moschitti. TANDA: Transfer and adapt pre-trained transformer models for answer sentence selection. In *Proc. of AAAI*, 2020.

[47] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune BERT for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer, 2019.

[48] Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. Aspect-based sentiment analysis using BERT. In *Proc. of NoDaLiDa*, 2019.

[49] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proc. of ACL*, 2018.

[50] Capuozzo Pasquale, Lauriola Ivano, Strapparava Carlo, Aiolli Fabio, and Sartori Giuseppe. Decop: A multilingual and multi-domain corpus for detecting deception in-typed text. In *Proc. of LREC*, 2020.

[51] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

[52] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[53] Brannvall Rickard, Ohman Johan, Kovacs Gyorgy, and Liwicki Marcus. Cross-encoded meta embedding towards transfer learning. In *Proc. of ESANN*, 2020. S.S. contribution.

[54] Mehryary Farrokh, Moen Hans, Salakoski Tapio, and Ginter Filip. Entity-pair embeddings for improving relation extraction in the biomedical domain. In *Proc. of ESANN*, 2020. Special Session contribution.

[55] Lauriola Ivano, Campese Stefano, Lavelli Alberto, Rinaldi Fabio, and Aiolli Fabio. Exploring the feature space of character-level embeddings. In *Proc. of ESANN*, 2020. Special Session contribution.

[56] Boes Wim, Van Rompaey Robbe, Verwimp Lyan, Pelemans Joris, Van Hamme Hugo, and Wambacq Patrick. On the long-term learning ability of LSTM LMs. In *Proc. of ESANN*, 2020. Special Session contribution.

[57] Kudashkina Katya, Wittek Peter, Kiros Jamie, and Taylor Graham W. Modular length control for sentence generation. In *Proc. of ESANN*, 2020. Special Session contribution.

[58] Worzyk Nils, Niewerth Stefan, and Kramer Oliver. Adversarials in speech recognition: Detection and defence. In *Proc. of ESANN*, 2020. Special Session contribution.