HAL open science

Machine Learning Meets Natural Language Processing - The Story so Far

Nikolaos-Ioannis Galanis, Panagiotis Vafiadis, Kostas-Gkouram Mirzaev, George A. Papakostas

To cite this version:

Nikolaos-Ioannis Galanis, Panagiotis Vafiadis, Kostas-Gkouram Mirzaev, George A. Papakostas. Machine Learning Meets Natural Language Processing - The Story so Far. 17th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI), Jun 2021, Hersonissos, Crete, Greece. pp.673-686, 10.1007/978-3-030-79150-6_53 . hal-03287671

HAL Id: hal-03287671

https://inria.hal.science/hal-03287671v1

Submitted on 15 Jul 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Machine Learning Meets Natural Language Processing - The story so far

Nikolaos-Ioannis Galanis[0000−0001−9528−4349], Panagiotis Vafiadis[0000−0002−6448−5524], Kostas-Gkouram Mirzaev[0000−0003−4473−4631], and George A. Papakostas[0000−0001−5545−1499]

HUman-MAchines INteraction Laboratory (HUMAIN-Lab), Department of Computer Science, International Hellenic University, Kavala, Greece
{nigaean, pavazei, gkmerza, gpapak}@cs.ihu.gr

Abstract. Natural Language Processing (NLP) has evolved significantly over the last decade. This paper highlights the most important milestones of this period, while trying to pinpoint the contribution of each individual model and algorithm to the overall progress. Furthermore, it focuses on issues still remaining to be solved, emphasizing on the groundbreaking proposals of Transformers, BERT, and all the similar attention-based models.

Keywords: Machine learning · Computational Linguistics· NLP· NLU· NLG· Linguistics· Ambiguity· CNN· BERT· Transformers· GPT

1 Introduction

Records of NLP application can be found even before the early 1900's when there were attempts of using machine translation to translate text from one language to another [12]. Meanwhile, there were some conflicting views between linguistics and computer science claiming that language is generative in nature and cannot be described with mathematical concepts [41].

Alan Turing adequately answered "Can machines think?" in 1950, by introducing the research/study of "Imitation Game" [33], a simulation process of a computer acting and answering without substantially changing the outcome[20]. Thus the machine is considered to be "thinking", as long as having a conversation with it could be indistinguishable from that with a human.

The first successful attempt to achieve that was ELIZA[56], a simple program within the Project of Mathematics and Computation ("Project MAC") at MIT that managed to mislead people into believing that it's a psychologist, reflecting on questions by turning the questions back at the speaker. Another program was PARRY (Colby, 1975) mimicking a paranoid schizophrenic [47]. Over the years, programs were getting "smarter" like Eugene Goostman [3] or Cleverbot [44] that statistically analyzes huge databases of real conversations to determine the best responses.

The downside was the inability to keep consistency and keep up with brand new subjects. There are numerous variations or alternatives to the Turing test,

like when humans have to prove their non-machine nature to a computer (ex. CAPTCHA [54]) or when we use AI to create original art.

1930 – Translating machines patents, Artsourni, Troyanskii propose dictionaries
1950 – Imitation Game by Alan Turing
1957 – Syntactic Structures by Noam Chomsky
1966 – ELIZA: Computer Psychotherapist by Joseph Weizenbaum
1968 – SHRDLU: NLU Computer Program by Terry Winograd (MIT)
1969 – Conceptual Dependency theory by Roger Schank
1975 – PARRY: Computer Schizophrenic person by Colby
1980 – ATN: Augmented Transition Network by William A. Woods
... – MUBBLE,MOPTRANS,KODIAK,ABSITY,DR.SPAITSO,RACTER
2006 – AI Software (Question Answering system) from IBM by Watson
2011 – Siri : Mobile Assistant by Apple

Fig. 1. A summarized timeline of important NLP milestones.

Over the last decade, NLP grew rapidly and led to next-gen applications, such as virtual assistants like Siri or Alexa. New methodologies were developed using neural networks or unsupervised learning for acquiring vector representations of words like Word2Vec or GloVe. The latest milestone in this growth, was the introduction of the attention-based models, using a mechanism that comprehends contextual associations between words and phrases.

NLP as a sub-field of AI, examines and detects patterns in data and uses them in achieving better understanding and generating natural language. There are several applications of NLP, some of them are:

1. Search engines
2. Virtual assistangs & Chatbots
3. Sentence segmentation
4. Part of speech tagging
5. Information Extraction
6. Question Answering
7. Machine Translation
8. Deep Analysis
9. Named entity recognition
10. Spam Detection
11. Text-to-speech & Speech-to-Text
12. Sentimental analysis
13. Text Summarization

NLP converts human language from the form of raw text data, into structured data (computer-understandable), but prior, it needs to perceive the data based on grammar, context and decide on intent and entities, with a process called Natural Language Understanding(NLU). On the other hand, Natural Language Generation(NLG), is a process that converts computer-generated data into human understandable text. This system generates well structured dynamic documents using both document-planning, micro-planning and realization, by representing human-like desired sentences[24].

Though other surveys have previously presented various trends in Natural Language Processing and Machine Learning ([21],[60],[38]), each in a different way either with a more practical or theoretical approach. In this review, we

will attempt to pinpoint and summarize the most critical and important breakthroughs in the field of NLP up until today, while also focusing on the still existent and new emerging challenges.

2 Materials and Methods

2.1 The Literature Accumulation

The search begins with Google Scholar's highest cited results for the last 10 years, while also tracing references and backwards citations. The search included at least one keyword of each group: Machine Learning (Machine Learning, Transformer, CNN, Neural Network, Recurrent, GRU, Deep Learning, Recursive, LSTM, ML) and NLP (Natural Language, NLP, NLU, NLG).

Clearly, there's a steadily increasing number of papers on the combined subject, booming after the proposal of Transformers, with publications of derived models and methodologies for transfer learning (see Fig. 2).

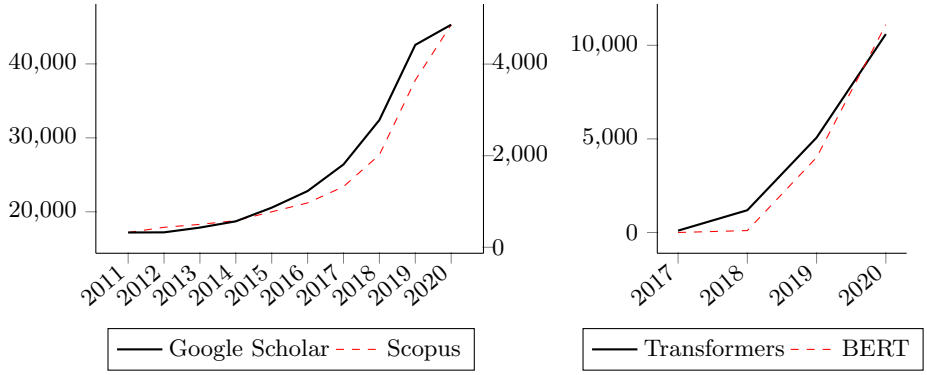

Fig. 2. Publications for NLP in general and references to popular Transformer papers.

2.2 The NLP field Transformation

Having a strong presence within the last decade, **Word embeddings** is a term where words that have the same meaning have a similar representation. In 2013, Mikolov introduced two different techniques for text vectorization: **Skip Gram** and **Common Bag Of Words** (CBOW) [37]. Both of them were released in a library under a single name, "**Word2Vec**" and later in the same year some improvements for both of them were suggested in an attempt to remedy polysemy [35]. Not long before, in 2011, the same author had also introduced a **Recurrent Neural Net Language Model**(RNNLM) being up to 15 times more efficient compared to past approaches [36].

In 2014 Pennington proposed an unsupervised learning algorithm for retrieving vector representations for words, named **GloVe** [39]. **Recurrent/Recursive Networks and LSTMs** [22] are intriguing recent developments in ML with

Sutskever suggesting the **Sequence to Sequence** (seq2seq) model[51]. Also at the same year, Kalchbrenner proposed a **Dynamic Convolutional Neural Network**(DCNN) [25] and Kim explored a variety of classification tasks [26].

Dong in 2015 introduced **multi-column convolutional neural networks** (MCCNNs) to analyze questions from multiple aspects and create their representations [19]. Yin presented a comparative study between CNN and RNN for NLP summing up the progress up until then [59]. Upadhyay in 2017 introduced a new method for managing polysemy in word embeddings [52]. Peters Suggested the Embeddings from Language Models in 2018 [40]. The Same year Chen introduced a new **LSTM** that outperformed all previous models on Natural Language Inference. [15].

2.2.1 Introducing Attention: the era of Transformers

Bahdanau, based on the previously proposed Encoder-Decoder architecture, introduced the term **"Attention"**[11]: an alignment score for each input word, based on the decoder's previous hidden state and the current input state of the sentence. Using this score, the decoder can decide which parts of the input sentence are the most important, without having to encode all of the input sentences into a fixed length vector.

Following this concept, Vaswani[53] made a rather bold proposal that leads to the **Transformers** architecture: the replacement of the costly RNNs with multi-headed self-attention layers in Encoder-Decoder models, thus increasing dramatically their performance, setting a new state-of-the-art for various tasks. Based on that idea, a whole new category of models emerged (Fig. 3). The **Transformer XL**[17] was suggested later on, attempting to resolve the limited length of the original Transformer's input.

Transfer learning and pre-training came along as another important progression. Howard and Ruder [23] proposed Universal Language Model Fine-tuning (**ULMFiT**) in 2018, a transfer learning method that could be applied to any task in NLP, consisting of 3 stages: training in a large amount of text to capture general features, fine tuning for the task at hand with discriminative fine tuning and slanted triangular learning rates, and finally adding and fine tuning the classifier layers. Discriminative fine tuning allows the tuning of each layer with a different learning rate, while slanted triangular learning rates linearly increases the learning rate initially, linearly reducing it then again afterwards.

By combining the idea of pre-training and separating the Encoder part of this new Transformer architecture and stacking it as many times as needed, OpenAI's team (Radford [42][43], Brown [13]) created 3 versions of the Generative Pre-Training model (a.k.a. **GPT-1/2/3**). Each version featured a larger number of parameters and pretraining in a larger corpus, achieving a new state-of-the-art for tasks like text generation and question answering with each version. The third version though is still not openly available, while a smaller model with 117 million parameters has been released for the second version.

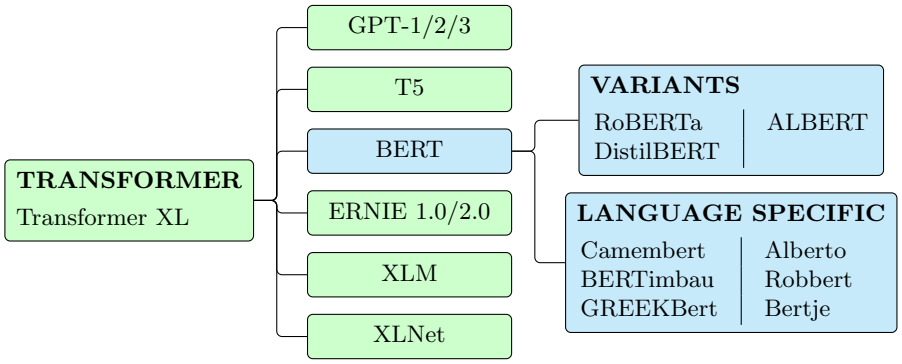

Fig. 3. The evolution of Transformer based models.

Bidirectional **E**ncoder **R**epresentations from **T**ransformers (BERT) technique was introduced by Devlin[18], with an architecture similar to that of GPT. As the name suggests, one of its basic differences with GPT is the bidirectionality that helps in better understanding of the context, giving it a crucial advantage over other models. By releasing a base and an (extremely) larger model, BERT achieved state-of-the-art performance in tasks like question answering and text classification and can be used for a variety of other NLP tasks just by fine-tuning with a much smaller task-specific corpus.

Various publications branched from the initial BERT release, attempting to improve it or provide a solution to its drawbacks. **RoBERTa**[31] was proposed as a better pretraining method, while **DistilBERT**[45] and **ALBERT**[29] were smaller, faster alternatives with training speed and reduced memory consumption in mind.

Yang proposed **XLNet** [58], an autoregressive (AR) model attempting to fix a discrepancy in the MLM task of BERT where the dependency between the masked input tokens is ignored. In order to achieve that, it's using a permutation language modeling objective – meaning that all tokens are predicted instead of only the 15% of BERT's masked tokens. And though AR models can usually access the context in one direction, the permutation allows it to be bi-directional. XLNet can outperform BERT – sometimes significantly – at 20 tasks.

Sun from Baidu introduced Enhanced Representation through kNowledge IntEgration (**ERNIE**) [49] at the beginning of 2019, a character-based model antagonizing BERT for the current state-of-the-art with a slight different masking strategy - multi-stage instead of the random one BERT has. Later, in 2020, Baidu released a second version of ERNIE [50] introducing "continual pretraining" and multiple training tasks for lexical, syntactical and semantical analysis, claiming to outperform BERT and XLNet not only for the Chinese language, but for English as well (see Table 1).

	Score	CoLA	SST-2	MRPC	STS-B	MNLI-m	QNLI	RTE	WNLI	AX
ERNIE	**90,9**	**74,4**	**97,8**	**93,9/91,8**	**93,0/92,6**	**91,9**	**97,3**	**92,0**	**95,9**	**51,7**
ALBERT	-	69.1	97.1	93.4/91.2	92.5/92.0	91.3	91.0	89.2	89.2	50.2
XLNet	-	70.2	97.1	92.9/90.5	93.0/92.6	90.9	-	88.5	92.5	48.4
RoBERTa	88.1	67.8	96.7	92.3/89.8	92.2/91.9	90.8	95.4	88.2	89.0	48.7
XLM	83.1	62.9	95.6	90.7/87.1	88.8/88.2	89.1	94.0	76.0	71.9	44.7
BERTlarge	80.5	60.5	94.9	89.3/85.4	87.6/86.5	86.7	92.7	70.1	65.1	39.6

Table 1: Results for various Transformers in the current GLUE Leaderboard [55].

2.3 More Data, Please!

The General Language Understanding Evaluation **(GLUE)** benchmark is a popular tool that evaluates the ability to analyze natural language understanding systems [55] featuring its very own leaderboard. A second version of it (**SuperGLUE**) came out a year later [46], featuring more and harder tasks, to achieve an even more accurate evaluation of the ever-evolving NLP models. GLUE consists of 11 tasks and their equivalent compilation of test datasets, while SuperGLUE features 10 more.

Common Crawl [2] is a repository with a significant amount of web crawl data, used as pre-training material for many of the models. It might be vast, but since it's a web dataset it's quite possible that heavy preprocessing is required before being actually usable. Other common sources of data between pretrained models are the Wikipedia pages - the main source for many multilingual and non-English models - as well as parsed subsets of the Reddit, IMDB, or Twitter websites. **Kaggle**[5] is another great source with a wide variety of user-submitted datasets, though due to their much smaller size they might be more suitable for fine tuning, rather than actually pretraining.

2.4 The Open Toolkits

	Stars	Description
spaCy [7]	19883	Python library with pretrained models and great multilingual support, not suggested for research and benchmarking.
Flair [4]	10091	PyTorch library developed by Zalando, featuring the "Flair embeddings" for more efficient text vectorization.
AllenNLP [1]	9794	Designed for quick prototyping and research with a variety of pretrained models.
NLTK [6]	9711	One of the most recognizable libraries for NLP with some ML options - though not a common option for the task.
Stanza [9]	5271	Stanford's Python library for "Many Human Languages". Can be used as an interface to CoreNLP for more features.
SparkNLP [8]	2002	Built on top of Apache Spark and TensorFlow for speed and scalability, with generic and domain specific models available.

Table 2: An overview of popular open source NLP tools & libraries.

In Table 2, a summary of popular open source NLP tools is presented. Github repository stars are used as an indicator of popularity, while also making sure the projects are still active with recent/frequent releases.

Furthermore, **Huggingface** [57] maintains a carefully curated git repository since 2019, with many of the latest pretrained Transformers for PyTorch and Tensorflow, allowing the quick testing of any of these models and turning prototyping into a breeze.

As one can easily see, there's a plethora of options - each with its own advantages and drawbacks. For instance, AllenNLP is more research/education oriented, while spaCy is probably a better choice for production, and NLTK might be harder to use. In the end, it all comes down to the application's requirements, and the developing team's personal preferences.

2.5 Challenges

Although NLP has evolved a lot over the last years, there are still some challenges. All the unstructured context needs to be translated into meaningful defined data in order to perceive the intended meaning and entities, based on grammar and context.

Text-Mining is used to identify non-trivial patterns in text-data, starting with the Data Collection, by building a corpus, Data pre-Processing handles and manipulates the corpus using sub-processes of tokenization, normalization and substitution. Most of the raw data are not useful to define features, usually containing a lot of noise, so the ML model tends to become less effective and difficult to train. The initial goal is to go from chunks with text to a list of cleaned tokens, and then proceed to data exploration & visualization, having a better dataset prior to building the model. Unstructured data transformed into useful text, by splitting the text into sentences, words and converted into standard form, like expanding constructions and set them to their base style.

The semantic meaning of words, ambiguity, grammar, or even slang is something that needs to be handled. In the next figure we are quoting some of the main challenges of NLP.

2.5.1 Ambiguity exists at every level in linguistics, as shown in Fig. 4. In natural language, it is common, words to have multiple meanings according the context of the sentence (contextual words). Contrariwise, different words can have the same meaning(synonyms). Irony, sarcasm and humor may use words with a specific state, but in fact imply the opposite. Some other main types of ambiguity are: Lexical ambiguity, where a single token can be presented as a verb, noun, or adjective, Semantic ambiguity refers to the conceptual situation described in a sentence having multiple interpretation[14] and Syntactic ambiguity is happening when there is a double meaning in a sentence and the syntax principles of the language are not followed. Additionally, errors, typos, slang and inconsistencies complicates the translation.[32] Many times in a sentence there is a discrepancy between the actual meaning and what is written. Pragmatics is the study dealing with this case.[30].

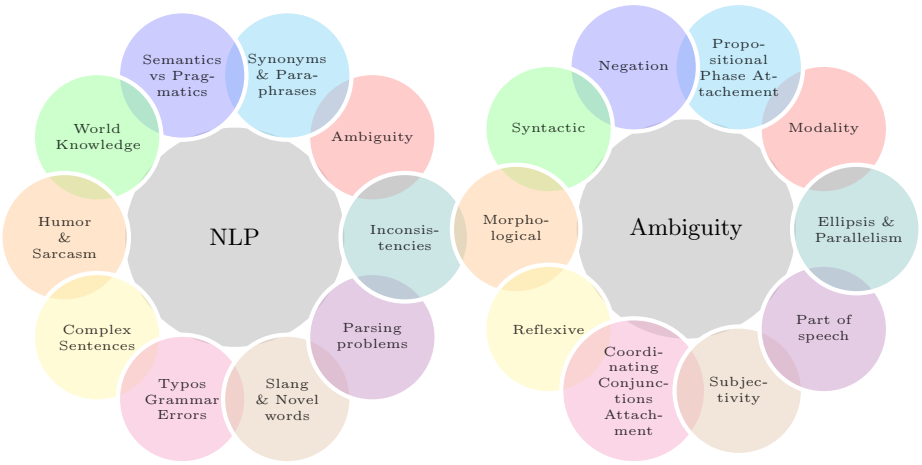

Fig. 4. NLP Linguistic Challenges while parsing human languages [10]

When input data is not text but speech, another ambiguity issue occurs. Phonetics and phonology refers to how tokens sound like. Phonetics deals with the vocal properties and perception, meaning how they are produced, and phonology deals with the expression in relation to each other in a language [27],[34].

2.5.2 Go Big (or maybe not?)

With every new publication, each team enters the race for a larger number of parameters (see Fig. 5), ending up with models with billions of parameters. While that might improve the actual results, it has a huge impact on the training cost - both financially and time-wise - even if just fine tuning is required, making many of the latest architectures unusable for single GPUs, TPUs or even whole clusters of such hardware - cloud based or local - in some cases. Besides whatever environmental consequences that might have, it's also **hindering their usage in real-world scenarios**.

According to Strubell's 2019 paper [48], training a BERT base model has 1,438lbs of CO_2 emissions, about 55 times more than these of the base Transformer model. Even worse, training a Transformer model with Neural Architecture Search emits 626,155lbs of CO_2, which is roughly equal to the lifetime emissions of 5 cars. And that training has an equally large cost when cloud services are used - starting at the whopping number of $942,973 in the case of NAS Transformer training scenario. So, it would probably be more efficient if researchers focused on the hyperparameters of each model and how they affect it's performance in order to improve them, than just adding more and more parameters.

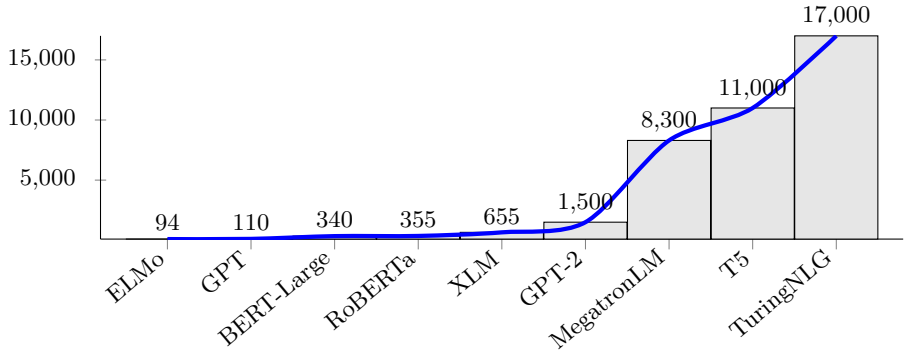

Fig. 5. Number of parameters for models over time (in millions) [45].

2.5.3 The Context Fragmentation & Text Repetition

The original Transformer architecture as well as popular deriving models has a predefined max input sequence length - in some cases defined by the model itself (e.g. 512 in BERT and GPT-1, 1024 for GPT-2) or by the overall hardware limitations (i.e. available memory). This might end up in **context loss in some marginal cases** where the input is split into segments where one's content correlates with its next or previous. Though in practice that's not usually a problem, there are some suggestions like the aforementioned Transformer XL that can help.

Furthermore, for the task of Question Answering (QA) or any other task that requires actual text generation, the output might be repetitive or even irrelevant on some occasions. Though sometimes that's due to the model itself or insufficient training, there are cases where the reason is this context fragmentation, or the lack of big/diverse enough datasets.

2.5.4 Lost in Translation

Most of today's pretrained models are focusing on the English language, with some approaches focusing on character based languages like Chinese or Arabic. Some of the very few multilingual options are sparse pretraining attempts of BERT or GPT for local languages with monolingual models as a result (see Fig.3). The multilingual version of BERT (or **m-BERT**), trained on the Wikipedia corpora, is another such option, supporting 104 different languages. The lack of a big enough, quality, multi-lingual dataset remains though, since even in Wikipedia many languages have a significantly lower number of pages (sometimes too low to even be considered) compared to English.

The **XLM model** [28] attempts to take multilinguality one step further by adding byte pair encoding and training BERT in two languages at once, while changing the "Masked Language Modeling" objective to "Translation Language Modeling", masking tokens in either of the two languages. While that creates

a multi-language model that can outperform m-BERT, it makes the data availability even harder, since now the same content is required in two languages at once. **XLM-R** [16] is an XLM successor, using a much larger dataset and improving performance at a scale that can be fine tuned for one language and then be used for cross-lingual tasks.

While all of the above do wield interesting results, there's still a huge gap between English and other languages in some tasks like QA. For instance, a F1/EM (Exact match) score of 80.6/67.8 in English becomes 68.5/53.6 in German for the MLQA question answering benchmark [16].

3 Conclusion

During the last decade there has been a tremendous progress in the field of NLP whether it's overall improvements, or task specific. It started with the vectorization revolution, with new suggestions for the most important task at the core of every NLP pipeline, ending up to the recent introduction of Transformers and transfer learning that marked the beginning of the "Golden Age".

Over this period, more and more companies started to adopt Chatbot technologies. Starting with simple pattern matching support agents, they're getting more and more sophisticated, gradually starting to adopt Machine Learning techniques and models, making them even more human-like and shifting to actual learning, away from patterns.

Machine Translation is also progressing steadily. Being one of the main tasks for the Encoder-Decoder model and their Transformer successors, automated translation applications are nowadays more accurate than ever. As the text generation models evolve, automated article generation applications start to emerge and virtual copywriters indistinguishable from humans, might soon be a reality.

But even though NLP has come this far, there are still many things to be done. Multilingual support models have still a long road ahead, and recent models will need heavy optimization and down-scaling or wider adoption of efficient hardware before they can be actually used in real-world applications. The Next Big Thing is still to be seen, and Transformers are currently leading the way.

Acknowledgements

This work was supported by the MPhil program "**Advanced Technologies in Informatics and Computers**", hosted by the Department of Computer Science, International Hellenic University, Kavala, Greece.

References

1. AllenNLP Github repository. https://github.com/allenai/allennlp, last accessed 21 Apr 2021
2. Common Crawl. https://commoncrawl.org/, last accessed 21 Apr 2021
3. Eugene goostman. http://eugenegoostman.elasticbeanstalk.com/, last accessed 21 Apr 2021

4. Flair Github repository. https://github.com/flairNLP/flair, last accessed 21 Apr 2021
5. Kaggle. https://www.kaggle.com/, last accessed 21 Apr 2021
6. NLTK Github repository. https://github.com/nltk/nltk, last accessed 21 Apr 2021
7. spaCy Github repository. https://github.com/explosion/spaCy, last accessed 21 Apr 2021
8. SparkNLP Github repository. https://github.com/JohnSnowLabs/spark-nlp, last accessed 21 Apr 2021
9. Stanza Github repository. https://github.com/stanfordnlp/stanza, last accessed 21 Apr 2021
10. Baeza-Yates, R.: Challenges in the interaction of information retrieval and natural language processing. In: International Conference on Intelligent Text Processing and Computational Linguistics. pp. 445–456. Springer (2004)
11. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
12. Brown, K.: Encyclopedia of language and linguistics, vol. 1. Elsevier (2005)
13. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020)
14. Bucaria, C.: Lexical and syntactic ambiguity as a source of humor: The case of newspaper headlines. Humor **17**(3), 279–309 (2004)
15. Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., Inkpen, D.: Enhanced lstm for natural language inference. arXiv preprint arXiv:1609.06038 (2016)
16. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116 (2019)
17. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-xl: Attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860 (2019)
18. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
19. Dong, L., Wei, F., Zhou, M., Xu, K.: Question answering over freebase with multi-column convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 260–269 (2015)
20. Epstein, R., Roberts, G., Beber, G.: Parsing the Turing test. Springer (2009)
21. Goldberg, Y.: A primer on neural network models for natural language processing. Journal of Artificial Intelligence Research **57**, 345–420 (2016)
22. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
23. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146 (2018)
24. Jurafsky, D., Manning, C.: Natural language processing. Instructor **212**(998), 3482 (2012)
25. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 (2014)
26. Kim, Y.: Convolutional neural networks for sentence classification (2014)
27. Kurdi, M.Z.: Natural language processing and computational linguistics: speech, morphology and syntax, vol. 1. John Wiley & Sons (2016)

28. Lample, G., Conneau, A.: Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291 (2019)
29. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942 (2019)
30. Leech, G.N.: Principles of pragmatics. Routledge (2016)
31. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
32. MacDonald, M.C., Pearlmutter, N.J., Seidenberg, M.S.: The lexical nature of syntactic ambiguity resolution. Psychological review **101**(4), 676 (1994)
33. Machinery, C.: Computing machinery and intelligence-am turing. Mind **59**(236), 433 (1950)
34. Mallamma, V.R., Hanumanthappa, M.: Semantical and syntactical analysis of nlp. International Journal of Computer Science and Information Technologies **5**(3), 3236–3238 (2014)
35. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
36. Mikolov, T., Kombrink, S., Burget, L., Černockỳ, J., Khudanpur, S.: Extensions of recurrent neural network language model. In: 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 5528–5531. IEEE (2011)
37. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. arXiv preprint arXiv:1310.4546 (2013)
38. Otter, D.W., Medina, J.R., Kalita, J.K.: A survey of the usages of deep learning for natural language processing. IEEE Transactions on Neural Networks and Learning Systems (2020)
39. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
40. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
41. Pullum, G.: Philosophy of linguistics. The Cambridge Companion to History of Philosophy **2015** (1945)
42. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
43. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
44. Saenz, A.: Cleverbot chat engine is learning from the internet to talk like a human. Singularity Hub (2010)
45. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
46. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4938–4947 (2020)
47. Saygin, A.P., Cicekli, I., Akman, V.: Turing test: 50 years later. Minds and machines **10**(4), 463–518 (2000)
48. Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in nlp. arXiv preprint arXiv:1906.02243 (2019)

49. Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., Wu, H.: Ernie: Enhanced representation through knowledge integration. arXiv preprint arXiv:1904.09223 (2019)
50. Sun, Y., Wang, S., Li, Y., Feng, S., Tian, H., Wu, H., Wang, H.: Ernie 2.0: A continual pre-training framework for language understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 8968–8975 (2020)
51. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. arXiv preprint arXiv:1409.3215 (2014)
52. Upadhyay, S., Chang, K.W., Taddy, M., Kalai, A., Zou, J.: Beyond bilingual: Multi-sense word embeddings using multilingual context. arXiv preprint arXiv:1706.08160 (2017)
53. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017)
54. Von Ahn, L., Blum, M., Hopper, N.J., Langford, J.: Captcha: Using hard ai problems for security. In: International conference on the theory and applications of cryptographic techniques. pp. 294–311. Springer (2003)
55. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: Glue: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461 (2018)
56. Weizenbaum, J.: Eliza—a computer program for the study of natural language communication between man and machine. Communications of the ACM **9**(1), 36–45 (1966)
57. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al.: Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771 (2019)
58. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. arXiv preprint arXiv:1906.08237 (2019)
59. Yin, W., Kann, K., Yu, M., Schütze, H.: Comparative study of cnn and rnn for natural language processing. arXiv preprint arXiv:1702.01923 (2017)
60. Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing. ieee Computational intelligenCe magazine **13**(3), 55–75 (2018)