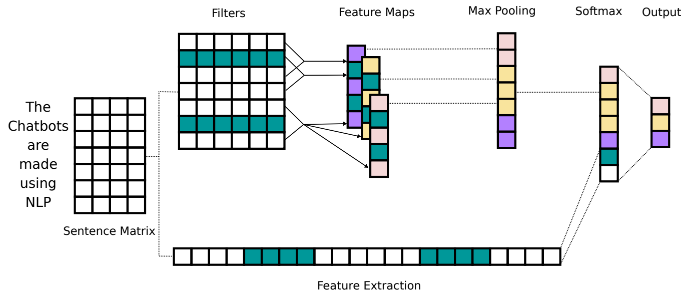
Graphical Abstract

A Brief Survey on Deep Learning based Natural Language Processing Systems

Jason Pandian, Dr. K. Krishneswari



Highlights

A Brief Survey on Deep Learning based Natural Language Processing Systems

Jason Pandian, Dr. K. Krishneswari

- This brief survey presents 5 phases of NLP history
- Some State-of-the–the-art Deep learning based NLP models are discussed

A Brief Survey on Deep Learning based Natural Language Processing Systems

Jason Pandian[a], Dr. K. Krishneswari[a]

[a]*Department of Computer Science and Engineering, Tamilnadu College of Engineering, Anna University, Coimbatore, 641659, Tamilnadu, India*

Abstract

Making a machine think and act as a human has always been a wild imagination of mankind for centuries. From the perspective of computing, Information Retrieval (IR) and Natural Language Processing (NLP) are the key technologies to address this issue to make our dreams come alive. Most of the human advances of this century were only due to the development of communication technologies that very much relied on IR and NLP. Nowadays, Natural Language Processing (NLP) plays a tremendous role in human life. In this survey, we tried to present a brief overview of 80 years of NLP history. Even though we outlined some of the classical, as well as modern NLP methods in this survey, we tried to give special attention to some of the state-of-the-art Deep Learning(DL) based NLP models that are giving new perspectives for Artificial Intelligence(AI) and other related technologies.

Keywords: NLP, IR, Machine Learning, Deep Learning, AI

1. Introduction

Deep learning models can be fit into various application domains and could produce state-of-the-art results. In recent years, Deep Learning models became a great tool for solving different kinds of natural languages problems. In this review, to provide a clear walk-through of deep learning-based NLP models and their evolution, we summarize, compare, visualize and highlight the various NLP methods as well as deep neural network models. We review deep learning models that can solve real-time problems and can be employed for numerous NLP tasks and try to understand the past, present and future trends of deep learning-based NLP systems.

1.1. Deep Learning (DL)

DL models are able to recognize and classify complex patterns in data using a deep, multilayered networks paradigm. It is often used for machine learning and artificial intelligence-related tasks. Some of the examples of deep learning algorithms are Deep Boltzmann Machine (DBM), Deep Belief Networks (DBN), and Convolutional Neural Networks (CNN). Generally, NLP models were designed using the CNN algorithm, so this paper will only address CNN related models.

1.2. Natural Language Processing(NLP)

NLP is one of the interesting and useful applications of Artificial Intelligence. In simple terms, an NLP system will try to mimic the ways of the human language understanding model to make sense of textual or speech content. Generally, this kind of NLP tasks will require the system to have the capability to translate, analyze and synthesize the language to make sense out of it.

Humans can effortlessly process both textual or speech content and understand them. The main task of an artificial NLP system is to replace the human aspect of understanding with a machine aspect of understanding and make the machine understand the natural language input and act accordingly. Mimicking the process of Understanding natural languages will be difficult to develop because of the numerous ambiguities and levels of contextual meaning involved in natural language. The inherent ambiguity of a language makes the NLP difficult to understand from a machine's perspective. There are five main categories into which language ambiguities fall: syntactic, lexical, semantic, referential and pragmatic[1]. These five aspects of language make it difficult to design a machine to act as a human to understand language content.

1.3. Overview of this Paper

This section provides a minimal introduction to NLP and DNN. The next section will present a brief history of NLP. Section III will explain Artificial Neural Networks and Deep Neural Networks in detail. Section IV will present some popular DL based NLP systems. Section V will present the comparison of the performance of those popular DL based NLP systems. And finally, section VI of the paper will conclude with a brief conclusion.

2. A Brief History of NLP

In [2], Karen Sparck Jones did a detailed review of NLP techniques from their origin in the late 1940s to the early 1990s. Karen's review identified and clearly distinguished four phases of NLP history based on the characteristics namely 1) emphasis on machine translation, 2) by the influence of artificial intelligence, 3) by the adoption of a logico-grammatical style, and 4) by an attack on massive language data. She defined the first phase of work in NLP as lasting from the late 1940s to the late 1960s, the second from the late 60s to the late 70s and the third from late 70s to the late 80s, and the fourth phase from the late 80s to the late 90s.

As shown in the previous paragraph, Karen's survey covered and classified almost 40 to 50 years of NLP history. Our survey extends her work by adding a fifth phase of NLP history which covers the remaining 30 years of NLP history until now; particularly, the deep learning related ones.

2.1. Phase I: Late 1940s to Late 1960s

Mostly the work of the first phase was focused on machine translation(MT). The following are some of the interesting aspects of the first phase pointed out by Karen.

- A MT task was done by translation by lookup using dictionary-based word-for-word processing.

- The computing resources available at that early period were very limited. The notable characteristics of that time were: The era of punched cards and batch processing. There were no suitable higher-level languages and programming was virtually all in assembly language. Access to computing machines was often restricted; they had very finite storage and were extremely slow.

- Even with the most advanced algorithms running on the best machines available at that time, for analysing long sentences, takes several minutes of processing time was needed which makes it impractical for implementing real-time MT systems.

- The practice of using computers for literacy and linguistic study began in this period, but none of them was closely linked with NLP at that time.

This phase was “syntax and semantics-based”. To make formal theories and concepts fit into hardware researchers were seeking for a novel computing hardware for data processing the NLP tasks. However, data processing itself was not well established in this early period of time.

2.2. Phase II: Late 1960s to Late 1970s

The second phase of NLP work was artificial intelligence (AI) flavoured one. The following are some of the interesting aspects of the second phase pointed out by Karen.

- It gave much more emphasis on world knowledge and on its role in the construction and manipulation of meaning representation.

- Mostly, the actual input to these systems was restricted and the language processing involved very simple compared with contemporary MT analysis.

- The latter works realized the need for inference on the knowledge base in interpreting and responding to language input.

- The need to identify the language user’s goals and plans was early recognised and has become a major trend in NLP research since, along with a more careful treatment of speech acts.

This phase was about AI-flavoured and semantics-oriented. The identification of language user's goals and plans become the major trend in this period of time.

2.3. Phase III: Late 1970s to Late 1980s

The third phase can be described as a grammatico-logical phase. The following are some of the interesting aspects of the third phase pointed out by Karen.

- This trend, as a response to the failures of practical system building, was stimulated by the development of grammatical theory among linguists during the 70s, and by the move towards the use of logic for knowledge representation and reasoning in AI.
- Computational grammar theory became a very active area of research linked with work on logics for meaning and knowledge representation that can deal with the language user's beliefs and intentions and can capture discourse features and functions like emphasis and theme, as well as indicate semantic case roles.
- The grammatico-logical approach was also influential in some other ways. It led to the widespread use of predicate calculus-style meaning representations, even where the processes delivering these were more informal than the purist would wish.
- The fourth trend of the 80s was a marked growth of work on the lexicon.

This phase was grammatico-logical. It satisfies the language user's beliefs and goals.

2.4. Phase IV: Late 1980s Onward

This phase was labelled as a massive data-bashing period and the following are some of the interesting aspects of the fourth phase pointed out by Karen.

- The rapid growth in the supply of machine-readable text has not only supplied NLP researchers with a source of data and a testbed for e.g., parsers.

- Use of text retrieval (cf. Jacobs, 1992) for document retrieval made this phase possible to process and available to various domains.

- This phase also concentrate on open question problems by the development of realistic NLP techniques like document retrieval and text parsers etc.

- This phase includes all the previous ideas of the past three phases: from syntax to semantics and semantics to grammatico-logical phases and grammatico-logical phases to lexicon(now).

- The development of some good processing systems as discussed in the third phase made user life easier. The users of this era benefited from the development of GUI for user experience. Hence, it was the major drawback of the last three phases.

By considering the negatives of the past three phases. This period has seen a significant, new interest in multi-modal, or multimedia systems. But whether combining language with other modes of media, like graphics, actually simplifies or complicates language processing was an open question until this fourth phase.

2.5. Phase V: Late 1990s to the Deep Learning Era

Even though the history of Deep Learning[9] can be traced back to 1943, the significant evolutionary step for DL took place only in 1999, when computers started becoming faster at processing data and graphics processing units (GPU) were developed[9].

Some of the important factors influencing the development of practical deep neural networks are :

- The developments towards Deep Learning was possible only because of the increase in memory, storage and computing capabilities of modern computers.
- It become possible because of the improvements in the capabilities of the Graphical Processing Unit (GPU)[9].
- It become possible because of the invention of Tensor Processing Units (TPUs)
- The research in this area developed very fast because of the open nature of research in DNN related technologies
- Another major factor that influenced this development was the "free" cloud computing resources generously provided by Google - this only promoted this research to a higher level

The following are the Noticeable milestones of DNN based models :

In 2001, a research report by META Group (now called Gartner)[9] described the increasing volume of data and the boosting speed of data as boosting the range of data sources and types. This was a call to prepare for the aggression of Big Data, which was just starting. This year is to be considered as the rise of Natural language models.

In 2008, two authors Collobert and Weston proposed a work on Multi-task learning[18]. It has been used successfully across all applications of machine learning from natural language processing. Collobert's and Weston's work was an eye-opener for speech recognition to computer vision.

In 2011, the acceleration of GPUs had increased significantly, making it possible to train convolutional neural networks "without" the layer-by-layer pre-training[9]. With the increased computing speed, it became clear Deep Learning had significant advantages in terms of efficiency and speed.

In 2013, Tomas Mikolov, et al.[19] from Google created a word embedding toolkit called as Word2vec. It can train vector space models faster than the previous approaches. The purpose and usefulness of Word2vec is to group the vectors of similar words together in vector-space. So it was considered as an eye-opener for novel DNN based NLP models.

In 2014, Ilya Sutskever, et al.[20] from Google created a deep neural network called as Sequence to sequence learning with neural networks. It was an Long short term memory(LSTM) model, that aims to map a fixed-length input with a fixed-length output where the length of the input and output may differ. So it performed well in language-based tasks.

In 2015, Dzmitry Bahdanau, et al.[21] proposed a machine translation based model, that is one of the core innovations in neural MT (NMT) and the root idea that enabled NMT models to outperform classic phrase-based MT systems and it was the best model to perform English to French Translation at that time.

In the same year 2015, the researchers came in different variants of approaches such as

- Alex Graves, Greg Wayne and Ivo Danihelka[22] created a deep neural network model called Neural Turing Machine. It was developed with a working memory, which gives it very impressive learning abilities. Unlike a standard neural network, it also interacts with a memory matrix using selective read and write operations.

- Jason Weston et al.[23] created a deep network model called Memory network, It combines learning strategies from the machine learning literature with a memory component that can be read and write.

- Sainbayar Sukhbaatar et al.[24] created a deep network model called End-to-end Memory Networks. It was developed with a recurrent attention model over a possibly large external memory. The architecture is a form of Memory Network, but it is trained end-to-end, and hence requires significantly less supervision during training.

- Kumar et al.[25] created a deep network model called Dynamic Memory Network(DMN). It processes input sequences and questions, forms episodic memories, and generates relevant answers. Due to its dynamic architecture, it was mainly optimised for question-answering(Q&A) problems and becomes a new path for NLP.

- Graves et al.[26] created a novel model called a differentiable neural computer (DNCs). DNCs was typically recurrent structure in its implementation, but it can learn like standard neural networks, but that can also store complex data like computers. DNCs indirectly takes inspiration from Von-Neumann architecture, making it likely to outperform conventional architectures in tasks.

- Henaff et al.[34] created a deep network model called Recurrent Entity Network (EntNet). EntNet was equipped with a dynamic long-term memory which allows it to maintain and update a representation of the state of the world as it receives new data.

These above-mentioned models help the DL-based NLP community to be developed in a wide area.

In 2016, Pranav Rajpurkar, et al. of Stanford University prepared a Q&A dataset. They call their work[28] SQuAD and it has organized more than 100,000 questions from Wikipedia articles. Using SQuAD, they evaluated the human performance of this QA task and posted it on the SQuAD website(with leader board)[29]. In SQuAD leader board, they posted their validated human performance on SQuAD in terms of F1-Score was 91.221.

In the same year, the Singapore Management University created a Long Short term memory network model and achieved an F1 score of 67.748[29], which was below human-level performance.

In 2017, the CMU created a model called Conductor-net and achieved F1 score of 81.415[29], it was a fine improvement within a year, and also it was near human-level performance. In the same year, Ashish Vaswani et al. at Google Brain proposed a deep transformer model based on attention mechanism[3]. It entirely changed the views of the researchers about the NLP. In the year 2018, Google AI Language released a deep bi-transformer model based on an attention mechanism called BERT[4]. It became famous and productive for different NLP tasks. In the year 2019, Facebook AI research published another BERT based model called RoBERTa[6]. It achieved an F1 score of 89.795 from [30]. This was near equal human-level performance. In the year 2020, another enhanced Albert based model named QIANXIN's SA-Net achieved an F1 score of 93.011[30]. This model outperformed humans.

Currently, in this decade, the processing of Big Data and the evolution of Artificial Intelligence are both dependent on Deep Neural Networks and Deep Learning techniques.

3. Deep Neural Networks (DNNs)

Traditional ANNs or simply NNs are simple mathematical structures generally denoting neurons with less number of hidden states.

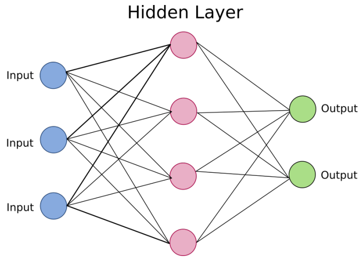


Figure 1: Block Diagram of Traditional Neural Networks.

The first traditional neural network or artificial neural network(ANN) was a Perceptron network created by Rosenblatt in 1958 and used pattern recognition with mathematical notation[11]. In 1974, Another model was proposed by Seppo Linnainmaa[12]. That model evaluates the gradient of the loss function with respect to the weights of the network for a single input-output. The idea of BPN was raised by a various scientist in the early 60's. But Linnainmaa was the first to propose and implemented to run on computers, that this approach could be used for the neural network after analyzing it in-depth in his 1974 dissertation. In the history of ANN, this same ideology with a change in the model may result in different NN architecture with respect to the computational power of the systems. Hence, these ANN can process entry-level tasks. When it comes to classical problem-solving. In the past, the Perceptron network and BPN were used to solve different types of image processing and classification tasks.

3.1. Deep Neural Networks(DNN)

DNN was inspired by the human brain and tries to mimic the functions of the human brain. A DNN is a network with a large number of hidden states. These hidden states (layers of neurons) can be used to process multidimensional inputs and can be tuned for complex classification tasks.

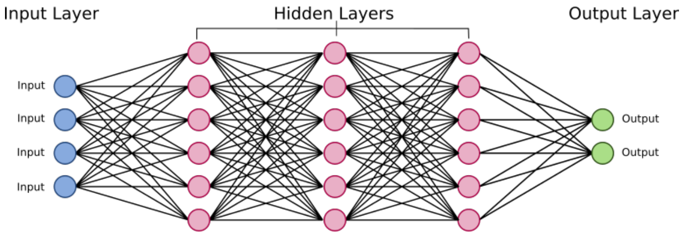


Figure 2: Block Diagram of Deep Neural Networks.

3.2. Recurrent Neural Network(RNN)

It was introduced in the 20th century by Psychologist David and Rumelhart. According to their work[14], the human biological brain functions the sequential data(biological data) by memorizing all past sequences. This principle is applied to the artificial neural network to memorize the data, the neural network is designed with the initial memory to process sequential data. In [31], the word "recurrent " means repetition, by processing the sequential data recurrently and storing its past sequences makes RNN works like a looped(feedback) network. Thus RNN can make an accurate prediction with temporal data-based applications.

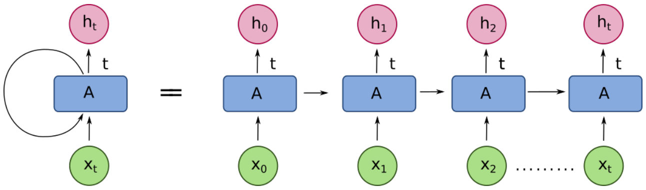


Figure 3: RNN Folded and Unfolded Version.

An RNN has four blocks,

- x(t) – input block, which gets the sequential data.
- A – the main block that contains the weight and activation function of the network.
- t – time-step, that holds the time required to process each sequence.
- h(t) – output block, the present output will be based on the previous outputs.

RNN has been the basic approach for training the sequential data and it has been the key to the development of new powerful models like LSTM and Gated Recurrent Unit(GRU) networks. The vanishing gradient problem was the major problem in RNN. From the perspective of NLP, an RNN can handle only small text sequences(RNN has short-term memory). It is difficult to train long sequences using normal RNN and also RNN may not be used for language translation. Further, training an RNN requires a large amount of cost and time. RNN is the foundation of different NLP models which are available today.

3.3. Convolutional Neural Network (ConvNet/CNN)

3.3.1. Biological Perspective of CNN

In [15], Hubel and Wiesel in the years the 1950s and 1960s proved that cat and monkey's visual cortexes contain neurons, that individually respond to small regions of the visual field without the movement of the eyes. The region of the visual space within which visual stimuli affect the firing of a single neuron is known as its receptive field. The Connecting cells have similar and overlapping receptive fields. The Receptive field size and location differ systematically across the cortex to form a complete map of visual space. The cortex in each hemisphere describes the contralateral visual field. Their work analyzes two basic visual cell types in the brain:

- Simple cells, whose output is maximized by straight edges having particular orientations within their receptive field.

- Complex cells, which have larger receptive fields, whose output is nonresponsive to the exact position of the edges in the field.

They proposed a cascading model of these two types of cells for using it in pattern recognition tasks.

In 1980, Kunihiko Fukushima introduced ”neocognitron”[16, 17]. His work explains the architecture of CNN. Even though it architecture was originally inspired by Hubel and Wiesel, which was analogous to that of the connectivity pattern of neurons in the visual cortex of the human brain. Individual neurons respond to stimuli only in a closed region of the visual field known as the receptive field. A collection of receptive field overlaps to cover the entire visual area. This principle used to develop CNN architecture.

3.3.2. Using CNN for a NLP Task

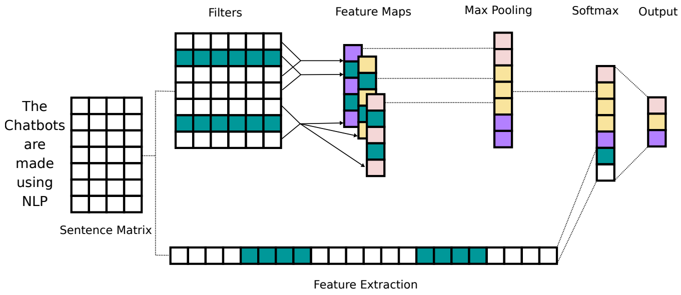


Figure 4: A CNN based NLP Task.

A CNN is a feed-forward deep learning network that passes inputs only in one direction, forward, from the first convolution layer, through all other convolution layers, and to the fully connected layer, and output is obtained expresses various aspects of the input[32]. A convolution is a process that transforms two functions into a new function. The pre-processing required in CNN is much lower than classical classification algorithms. While in primitive methods, filters are hand-engineered. On the other hand, CNN has the ability to learn these filters/characteristics by itself.

That means CNN needs no manual pre-processing. The CNN filters are not defined instead the value of every filter is learned during the training process itself. CNN executes hierarchical feature learning. CNN does not encode the position and orientation of objects. This makes convolution operations well carried out in low-end machines. Such as smartphones, embedded systems, and IoT devices. The modern CNN is one of the main categories to do image recognition, image classification, as well as considered the key for the development of almost all the novel NLP models that are good in running real-time applications that are available today.

3.4. Difference Between Neural Networks and Deep Neural Networks

Deep Neural Networks is nothing but a very large traditional NN, which will typically need higher-end hardware to make real use of it. In Table 1, we present some of the differences between traditional neural networks and deep neural networks.

4. Deep-Learning-Based NLP Systems

Deep Learning is not a novel field. The first Deep Learning model was published by Alexey Ivakhnenko and Lapa in 1967. This paper[13] described a deep network with eight layers trained by the group method of data handling, while the algorithm worked, training required by this algorithm took 3 days. It is clear that Deep Learning requires high computation power.

By definition, it is a family member of machine learning that can automatically extract the features from the data while processing large datasets. It was often interchanged with the word deep neural networks or artificial neural network. Hence, nothing was changed comparing the late 60s and 70s, except the extreme computation power and parallelization techniques of 21st-century computing machines.

In this section, we present some of the popular and most widely used deep learning models in recent years. The models described here are widely used in the design of NLP systems. So these models are called deep language models. We also focus on these models based question answering systems(QAS). The one commonality in all the discussed models is: all of them are 'transformers' based on the attention mechanism.

4.1. Bidirectional Encoder Representations from Transformers (BERT)

BERT was a deep neural network(language model) created by Jacob Devlin, et al. from Google[4]. It was made up of transformers, and stack multiple transformer encoders on top of each. It uses bidirectional learning as opposed to directional models. BERT tries to understand the whole context by addressing each word from left and right. BERT architecture is based on Self-Head Attention. BERT uses two pre-training strategies: masked language modelling(MLM) and the next sentence prediction (NSP). BERT was readily pretrained using these two pre-training strategies on the BookCorpus (800 million words) and English Wikipedia (2500 million words). This makes pretrained BERT distinguished from past DNN models. It can readily be used in medium-cost machines. By transfer learning, BERT can be modified into various NLP tasks by adding one or a few core layers in its

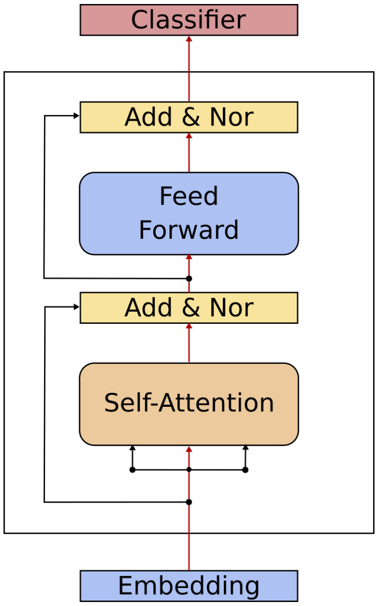


Figure 5: BERT Architecture with Self-attention.

end. This process is called fine-tuning. These salient features of the BERT made NLP enthusiasts personalize, develop, evolve, research and optimize new models.

The“Fig. 5” shows the architecture of BERT. As shown in this figure, the self-attention block was responsible for possible word prediction that was described in [4]. Other blocks are common functional blocks found almost in every transformer-based architectures.

4.2. XLNet

XLNet was a deep neural network(language model) created by Zhilin Yang et al. from Google AI Brain Team[5]. It was a large bidirectional transformer, that was improved the training process with larger data and use more computational power to achieve better than BERT prediction metrics

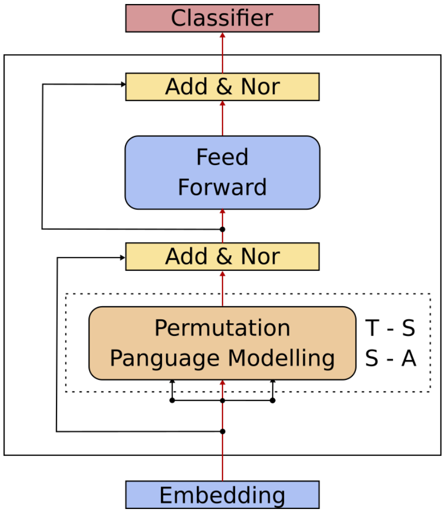


Figure 6: XLNet Architecture with Two-stream Self-attention(T - S S - A)

on 20 language tasks. To improve the training, It introduces permutation language modelling, where all tokens are predicted but in random order. Its architecture was based on Two-stream Self-attention. This contrasts with BERT's masked language model with self-attention, where only the masked (15%) tokens are predicted. This was also in contrast to the other language models, where all tokens were predicted in sequential order instead of random order. This helps the model to learn efficient bidirectional relationships and therefore better handle the dependencies and relations between words. In inclusion, Transformer XL was used as the base architecture, which produced good performance even in the absence of permutation-based training.

The "Fig. 6" shows the architecture of XLNet, As shown in this figure, the permutation language modelling block was responsible for random order predictions that were described in [5]. Other blocks are common functional blocks found almost in every transformer-based architectures.

4.3. Robustly-optimized BERT approach (RoBERTa)

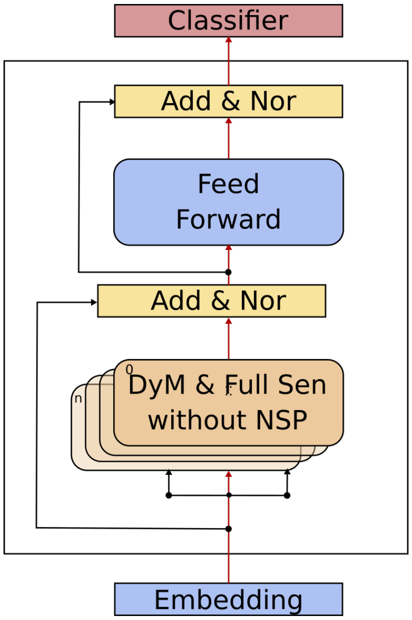


Figure 7: RoBERTa Architecture with Dynamic Modeling.

RoBERTa was a deep neural network(language model) created by Yinhan Liu et al. of Facebook[6]. It outperformed both XLNET and BERT in the Glue benchmark. It improved the BERT masked language modelling with a strategy where they remove Next Sentence Prediction in BERT and introduced an idea called dynamic masking. Hence that masked token changes throughout each epoch, which made the model learn to predict intentionally hidden secrets of text. They also improved the hyper-parameters tuning for the BERT and trained the BERT model using much larger mini-batches and learning rates.

The“Fig. 7” shows the architecture of Roberta. As shown in this figure, the Dynamic modelling and Full Sentence without NSP(DyM & Full sen without NSP) block is responsible for possible word predictions that were described in [6]. Other blocks are common functional blocks found almost in every transformer-based architectures.

4.4. DistilBERT

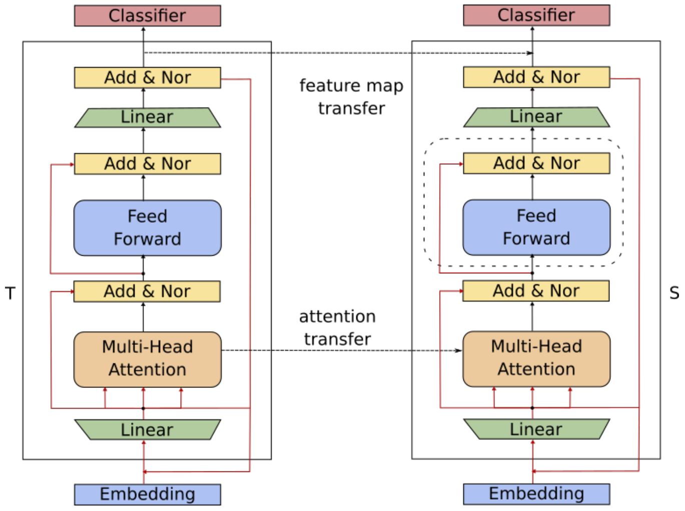


Figure 8: The Teacher-Student Architecture of DistilBERT

DistilBERT was a deep neural network(language model) created by the NLP enthusiasts Victor SANH et al. at Hugging Face[7]. It showed that it is possible to reach and achieve 97% of BERT's language understanding capabilities while reducing the size of the BERT model by 40%. Moreover, this model was 60% faster. While relying on the BERT architecture and using the same training data, DistilBERT removed the token-type embeddings and pooler (which BERT uses for the next sentence classification task) and also implemented a few ideas from RoBERTa and used a knowledge distillation process for the training of the model.

The“Fig. 8” shows the architecture of DistilBERT. As shown in this figure, the T & S describes the teacher-student model, DistilBERT - student(S) can perform up to 97%n of the knowledge gained from the 100%n BERT - teacher(t), even student was reduced to 40%, it can perform 60% faster. So DistilBERT can be used for low cost based NLP tasks.

The term “DistilBERT” is derived from the distillation process from science, which is a process of separating water and salt. In NLP, distillation refers to knowledge distillation which means to train a student model (DistilBERT) based on an already trained teacher model (BERT). In this case, DistilBERT is trained to copy the behaviour of BERT by equaling the output distribution - training through knowledge transfer.

4.5. A Lite BERT (ALBERT)

ALBERT was a deep neural network(language model) created by Zhenzhong Lan1 Mingda Chen et al. from Google Research collaboration with Toyota Technological Institute[8]. They pointed out that, the RoBERTa focused on performance and DistilBERT on speed, ALBERT is built to address both. Their model achieved better results with lower memory consumption and increased training speed compared to BERT. They also claim that BERT was parameter inefficient and apply techniques to reduce the parameters to 1/10th of the original model without substantial performance loss. Building on the BERT architecture, they demonstrate two strategies to reduce the model size: factorized embedding parameterization and cross-layer parameter sharing. In inclusion, they improved the model training by changing the next-sentence-prediction task for sentence-order prediction by keeping the equal amount of training data as BERT and DistilBERT.

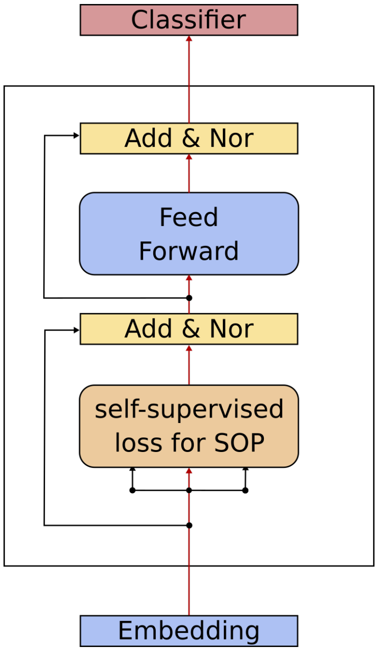


Figure 9: ALBERT Architecture with SOP

The “Fig. 10” shows the architecture of ALBERT. As shown in this figure, the “self-supervised loss for Sentence-order Prediction(SOP) block” was responsible for possible word predictions that were described in [8]. Other blocks are common functional blocks found almost in every transformer-based architectures.

5. Comparison of Different DNN Models and Their Performance

In this section, we present some of the important parameters and the performance scores of five deep neural network models, BERT, XL-Net, RoBERT, DistilBERT and ALBERT. The results of the performance were arrived on the popular Question Answering Dataset called SQuAD.

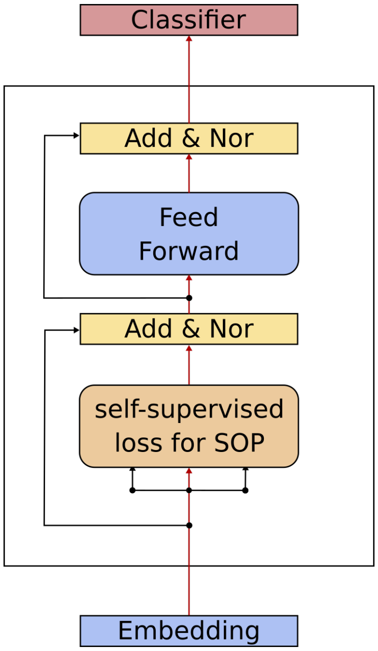


Figure 10: ALBERT Architecture with SOP

5.1. *Stanford Question Answering Dataset (SQuAD)*

5.1.1. *SQuAD1*

We used some of the results using data set SQuAD which was published by Pranav Rajpurkar et al. [27],[28],[28]. It was a reading comprehension dataset, consisting of more than 100,000 questions posed by crowd-workers on a set of Wikipedia articles, where the answer to every question is a segment of text or span from the corresponding reading passage, or in some cases the question might be unanswerable.

5.2. *Comparison of Single Model Performance With SQUAD2.0 dataset*

In the following table 2, we present the performance of those five deep neural network models in terms of EM score and F1 score.

Except for the results of DistilBERT, all others were from the SQUAD2.0 test dataset. These results were obtained from the SQUAD2.0 leaderboard. The result of DistilBERT is from [10] and this is the scores were originally obtained with SQUAD2.0 DEV dataset.

If we carefully study the Table 3 and Table 2, we can understand the outstanding performance of the ALBERT model. Because the models XLNet, RoBERTa have high parameters in nature, their performance is low. On the other hand, ALBERT with less number of parameters in nature and the fine-tuned ALBERT on SQuAD2.0 outperform other models and as well as human performance. And also ALBERT was able to do it in a very fast way with less computational resources. From those tables, we can understand that the optimization of models by understanding problems may help to build a small and super-efficient model in future.

5.3. Parameters of Different DNNs

In Table 3, we present some of the important parameters of five deep neural network models. These parameters describe some of the aspects of practical implementation of these deep neural networks.

For better understanding, we present the following bar chart (Fig. 11) for clear visualization of the differences in EM Score of the compared models.

For better understanding, we present another bar chart (Fig. 12) for clearly visualization the differences in the F1 score of the compared models.

5.4. Observations and Findings

As we did this survey, we realized the important improvements in NLP techniques of the different phases of NLP history. Based on that observation,

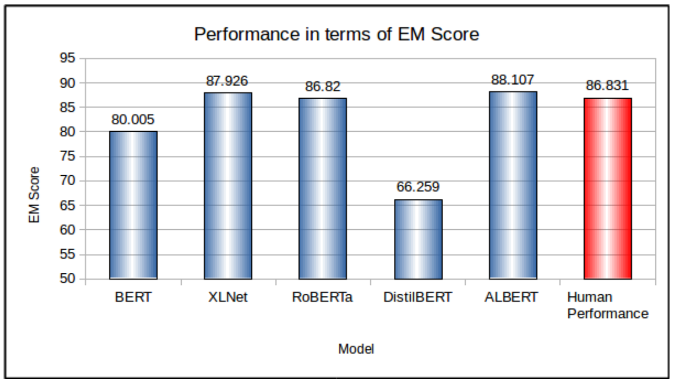


Figure 11: Performance of the Models in terms of EM-Score

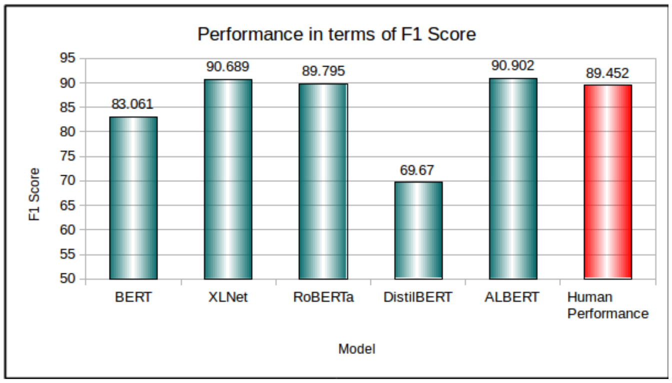


Figure 12: Performance of the Models in terms of F1-Score

with respect to performance, we can classify all algorithms of NLP into five classes as algorithms that are giving:

- Below Human-level Performance
- Near Human-level Performance
- Equal Human-level Performance
- Above Human-level Performance and

- Super/Hyper Human level of Performance

If we carefully study the five phases of NLP presented in our Brief History of NLP in section 2, it is obvious that almost all algorithms or models invented up to the first four phases of NLP history were only achieved 'Below Human-level Performance'. Only during the fifth phase of NLP history, the algorithms and models achieved 'Near Human-level Performance', 'Equal Human-level Performance', and even 'Above Human-level Performance'. And DNNs played an important role in achieving the first four levels of performance. In the same ratio of growth of DNN and NLP, achieving a Super/Hyper Human level of performance will also be a possibility in near future.

6. Conclusion

In this work, we did a brief survey on five historical timelines of the NLP era. Further, we did a study on five transformer-based BERT models by analyzing, visualizing, and comparing them with one another.

As we observed in our survey, DL based NLP models performed poorly even until 2017 and their results are very lower than human-level performance. After that, with the availability of heavily powered machines, DL based NLP models started to perform good, but their performance was still a little bit low comparing with actual human performance. During 2018, some state-of-the-art models evolved and played a huge role in NLP tasks and achieved equal to human performance. After that, Multi powered(layered or ensembled) state-of-the-art models evolved and started to achieve above-human level performance in various NLP tasks. As we mentioned in the previous section, the NLP model based on Deep Learning is quite interesting and was able to achieve an above-human level of performance.

Today, DNN based NLP systems capable to translate any language text or speech to any other known language. In the future, NLP may become superintelligent and there may be a possibility of fine-tuned NLP systems that may decipher the ancient, unknown, language scriptures and hence replace the traditional language expert. Future NLP model may provide human-like or better human-like Bots/Chatbots that can be your future caretaker, companion or next-generation virtual teacher. The future NLP model may show a possibility to communicate with aliens(if there will be something) that often fantasize in best selling science-fiction, and fiction emulated movies. By surviving the past and present of NLP and DNN, this survey ends with an endnote that one-day humans may achieve DNN/NLP/AI systems that can be superintelligent and can be optimized to solve the present unsolved problems of Quantum Physics and other mathematical secrets of nature.

References

[1] Finlay, Janet, and Alan Dix, “An introduction to Artificial Intelligence”, UCL Press, London, 1996

[2] Karen Sparck Jones, “Natural Language Processing: A Historical Review”, Computational Linguistics, vol. 9-10; Pisa, Dordrecht, 1994.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, “Attention Is All You Need”, Google Brain, arXiv:1706.03762v5 [cs.CL] 6 Dec 2017.

[4] Jacob Devlin, Ming-Wei, Chang Kenton Lee, Kristina Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, Google AI Language, arXiv:1810.04805v2 [cs.CL] 24 May 2019.

[5] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le,“XLNet: Generalized Autoregressive Pretraining for Language Understanding” , Carnegie Mellon University, Google AI Brain Team, rXiv:1906.08237v2 [cs.CL] 2 Jan 2020.

[6] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, Paul G., “RoBERTa: A Robustly Optimized BERT Pretraining Approach”, arXiv:1907.11692v1 [cs.CL], 26 Jul 2019.

[7] Victor SANH, Lysandre DEBUT, Julien CHAUMOND, Thomas WOLF, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”, arXiv:1910.01108v4 [cs.CL] 1 Mar 2020.

[8] Zhenzhong Lan1 Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut, “ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE REPRESENTATIONS”, A conference paper at ICLR 2020, arXiv:1909.11942v6 [cs.CL] 9 Feb 2020.

[9] Keith D. Foote, “A Brief History of Deep Learning”, 2017. https://www.dataversity.net/brief-history-deep-learning/

[10] , “Evaluating QA: Metrics, Predictions, and the Null Response”, Cloudera Fast Forward Labs, Minneapolis, MN, Jul, 2020.

[11] Rosenblatt F, “The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain”, Psychological Review Vol. 65, No. 6,1958.

[12] Linnainmaa, Seppo, “The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors”, Univ. Helsinki, 1970

[13] Ivakhnenko, A. G., Lapa, V. G., “Cybernetics and Forecasting Techniques”, American Elsevier Publishing Co. ISBN 978-0-444-00020-0, 1967.

[14] Williams, Ronald J.; Hinton, Geoffrey E., Rumelhart, David E. , “Learning representations by back-propagating errors”. Nature. 323 (6088): 533-536, ISSN 1476-4687. October 1986.

[15] David H. Hubel and Torsten N. Wiesel, “Brain and visual perception: the story of a 25-year collaboration”, Oxford University Press US. p. 106. ISBN 978-0-19-517618-6, 2005.

[16] Fukushima, K., “Neocognitron”, doi:10.4249/scholarpedia.1717, 2007.

[17] Fukushima, Kunihiko, “Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position”, Biological Cybernetics, 1980.

[18] Collobert, Weston, “Machine Learning, Proceedings of the Twenty-Fifth International Conference”, Univ.Helsinki, Finland, 2008.

[19] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, “Efficient Estimation of Word Representations in Vector Space”. ArXiv:1301.3781, 2013.

[20] Ilya Sutskever, Oriol Vinyals, Quoc V. Le, “Sequence to sequence learning with neural networks”, ArXiv:1409.3215, 2014.

[21] Dzmitry Bahdanau, Cho Kyunghyun, Yoshua Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate”, ArXiv:1409.0473, 2014.

[22] Alex Graves, Greg Wayne, Ivo Danihelka, “Neural Turing Machines”, Google DeepMind, arXiv:1410.5401v2 [cs.NE], 2014.

[23] Jason Weston, Sumit Chopra, Antoine Bordes, “Memory Networks”, Facebook AI Research, arXiv:1410.3916v11 [cs.AI], 2015.

[24] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, Rob Fergus, “End-To-End Memory Networks”, New York University, arXiv:1503.08895v5 [cs.NE], 2015.

[25] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, Richard Socher, “Ask Me Anything: Dynamic Memory Networks for Natural Language Processing”, CA USA, arXiv:1506.07285v5 [cs.CL], 2016.

[26] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu and Demis Hassabis , "Hybrid computing using a neural network", 2016.

[27] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, Percy Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text", Stanford University, arXiv:1606.05250v3 [cs.CL], 11 Oct 2016.

[28] Pranav Rajpurkar, Robin Jia, Percy Liang, "Know What You Don't Know: Unanswerable Questions for SQuAD", Stanford University, arXiv:1806.03822v1 [cs.CL], 11 Jun 2018.

[29] SquAD(closed), https://rajpurkar.github.io/SQuAD-explorer/explore/1.1

[30] SquAD2.0, https://rajpurkar.github.io/SQuAD-explorer/

[31] Jason Pandian, "Understanding Recurrent Neural Network", A Tech Blog, https://jason.co.in/understanding-recurrent-neural-network/

[32] Jason Pandian, "Understanding Convolutional Neural Network", A Tech Blog, https://jason.co.in/understanding-convolutional-neural-network/

[33] Jason Pandian, "Difference Between Traditional Neural Network and Deep Neural Network", A Tech Blog, https://jason.co.in/difference-between-traditional-neural-network-and-deep-neural-network/

[34] Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, Yann LeCun, “Tracking the World State with Recurrent Entity Networks”, arXiv:1612.03969v3 [cs.CL], 10 May 2017.

Table 1: Difference Between NN and DNN [33]

	Traditional Neural Networks	**Deep Neural Networks (DNN)**
Hardware	Traditional Neural Networks require only Lower-end processors (i.e, it can function with less number of Central Processing Unit Cores) and Lower-end Graphics Processing Units. It also requires less power supply	DNN requires Higher-end processors (ie, it can function only with a high number of Central Processing Unit Cores), Higher-end Graphics Processing Unit and Tensor Processing Unit. It also requires a large amount of power supply.
Training Time	Traditional NN can be easily trained and executed in low-end machines. It requires less amount of time to train a typical neural network	DNN can only be trained on higher-end hardware and will require several hours or even several days of CPU/GPU/TPU time.
Training Algorithm	Traditional Simple training algorithms such as SGD will be sufficient to train a Traditional NN for a typical problem.	The complexity and vastness of DNN problem space will require the best algorithms (Adam and its variants are believed to be providing better performance on DNN.) Block Diagram of Traditional Neural Networks
Epochs	A small problem related to Traditional NN typically will require several epochs for achieving better performance. Further improvement of training will linearly increase with respect to the training epochs. So hopefully we will have a better-trained model at the final epoch.	In most of the NLP problems, generally, the performance of the training will not get improved with respect to the epochs as in the case of NN. However, it will achieve acceptable accuracy even in a few epochs. However, most of the time, the model belonging to the final epoch will not be the better model. Even a model belonging to a lesser epoch may perform well. So as a practice, people will choose a better model from N epochs of training.
Parameters	A typical Traditional NN and a related problem will generally need less than a hundred or a few hundred parameters during training.	On the other hand, a typical DNN generally needs millions of parameters for solving problems such as NLP Applications.
Example	SVM, RBF, etc., are the most successful Traditional NN models	There are a lot of Successful models with respect to the area of their application.(LSTM, Bi-LSTM, CNN, BERT, You only look once(YOLO), etc)

Table 2: Performance with Stanford Question Answering Data-set - SQUAD2.0

Method	**Date**	**Performance**	
		EM	**F1 Score**
BERT (Single model) by Google AI Language	Nov 09, 2018	80.005	83.061
XLNet (Single model) by Google Brain & CMU	Nov 15, 2019	87.926	90.689
RoBERTa (Single model) by Facebook AI	Jul 20, 2019	86.820	89.795
DistilBERT[10] (Single model) [a]	-	66.259	69.670
ALBERT (Single model) by Google Research & TTIC	Sep 16, 2019	88.107	90.902
Human Performance by Rajpurkar & Jia et al. Stanford University	2018	86.831	89.452

[a] This result is from[10].

Table 3: Comparison of the State of the Art Models

Parameter	**BERT**	**XLNet**	**RoBERTa**	**DistilBERT**	**ALBERT**
Model Date	Oct 11, 2018	May 21, 2019	May 26, 2019	Oct 2, 2019	Sep 26, 2019
Method	Bidirectional Transformer, MLM & NSP	Bidirectional Transformer with Permutation based Modeling	BERT Without NSP using Dynamic Masking	BERT Distillation	BERT with Reduced Parameters & SOP
Layers/ Hidden/ Attention Heads	Base: 12/768/12 Large: 24/1024/16	Base: 24/1024/16	Base: 12/768/12 Large: 24/1024/16	Base: 6/768/12	Base: 12/768/12 Large: 24/1024/16
Parameters (in Millions)	Base:110 Large:340	Base: 110 Large: 340	Base:125 Large:355	Base:66	Base:12 Large:18
Training Data	BookCorpus+ English Wikipedia =16GB	Base: 16GB BERT Large: 16GB BERT+ 97GB Additional =113GB	16GB BERT+ CCNews + OpenWebText +Stories =160GB	BookCorpus+ English Wikipedia =16GB	BookCorpus+ English Wikipedia =16GB
Training Time	Base:8 x V100 x 12d Large:280xV100 x1d	512TPU x 2.5days (5 times more than BERT)	1024 x V100 x 1d (4-5 times higher than BERT)	8 x V100 x 3.5d	Base: - Large: 1.7 Times faster
Performance	Outperform all models of Oct 2018	2-15 % Higher Performance over BERT	88.5 on GLUE	97% of BERT base's Performance on GLUE	89.4 on GLUE