Deep Learning for Applications in Natural Language Processing: A Survey

Md Muhtasim Billah, School of Mechanical and Materials Engineering
Washington State University, Pullman, WA 99163

Abstract— In the last decade, scientific research on natural language processing has been significantly influenced by an eruption in the use of deep learning techniques. This survey aims at presenting a brief introduction of this field as well as a swift overview of deep learning architectures, existing methods for natural language processing and how the latter is being benefitted from the former. It then explores the plethora of recent works and recaps a variety of relevant literatures which put emphasis on the core linguistic processing issues as well as applications in scientific fields. The survey ends with a discussion on the prevailing state of the art along with recommendations for future research in the field.

***Index Terms*—natural language processing (NLP), deep neural network (DNN).**

I. Introduction

THE field of natural language processing (NLP) embodies a variety of topics that involve the computational processing and understanding of human languages. Beginning in the 1980s, the field's dependency on data-driven approaches involving statistics, probability, and machine learning [1], [2] has been increased. But the recent surge in computational capabilities, harnessed by elevated parallel processing platforms that utilizes graphical processing units (GPUs) [3], [4] has paved the way for *deep learning*, which exploits artificial neural networks (ANNs), containing millions and sometimes billions of trainable parameters [5]. On top of that, the availability of massive datasets, often called *big data*, facilitated by sophisticated data collection procedures, enables the training of such deep architectures [6], [7], [8] and making highly accurate predictions.

In recent years, academic researchers and industrial developers in the field of NLP have been able to benefit from the power of modern ANNs with intriguing results. Specially, in the last few years, deep neural networks (DNN) have upsurged remarkably [9], [10] leading to significant improvements both in core NLP areas as well as in the more practical, industrial grounds. This survey first provides a concise overview of both NLP and DNN, and then offers a discussion on how deep learning is being used in NLP. The emphasis will more be on discussing the applications of deep learning in computational linguistics rather than elaborating on the underlying theories. The discussion on the recent progresses in the field should be useful to familiarize oneself with the current state of the art before diving into the advanced research.

The article has been carefully divided into several sections. To present a relevant background, the topics of NLP and DNN are introduced in Section II. The motivation for using deep learning for NLP is explained in Section III. The ways in which deep learning has been used to solve problems in core areas of NLP are presented in Section IV. Some light has been shed on the promising future in this regime in Section V. Conclusions are then drawn in Section VI with a brief summary of the predictions, suggestions, and other thoughts on the future of this dynamically evolving area.

II. Relevant Background

Advancement in the computational power and the ability to collect and manage a humongous amount of data has paved the way for rejuvenating the possibilities of artificial intelligence. The core subset of AI is machine learning (ML) which includes a certain category of complex non-linear algorithms called the deep neural network (DNN) that lies at the center of deep learning (DL). The progress in ML, like any other scientific fields, has also facilitated the regime of natural language processing (NLP) and its applications in numerous sectors. Before diving into the matter of how DL has influences NLP, a brief relevant background will be offered in this section of the article.

A. Deep Learning (DL)

By definition, deep learning refers to that specific field of supervised machine learning that facilitates the use of a deep neural networks to train on a large volume of labeled dataset to learn its relevant parameters such as the weights and biases via a suitable choice of optimization algorithm with systematically tuned hyperparameters. Where a deep neural network differs from a shallow neural network is that it has two or more hidden layers in its architecture. Although the research in neural net was launched decades ago, it has boomed in the last decade and numerous variants and improvements of the basic neural network have been developed. The most common and widely used deep neural network applications are discussed below.

i. Multi-Layer Perceptron (MLP)

Multi-layer perceptron or MLP is the simplest type of feed forward neural network (FFNN). A basic MLP architecture contains at least three layers which are the input layer, hidden layer and the output layer. In each layer, there are multiple neurons where non-linear activation functions are deployed. The neurons of the same layer are not connected with one other; however, each neuron of a layer is connected to each neuron of the following layer. Each of the neurons in the input layer is called an input feature and there are randomly initialized wights and biases associated with them. The linear combinations of these features, weights and biases are then passed through a non-linear activation function in each of the neurons of the following hidden layer. The output of the activation function in the hidden layer then works as the input features for that layer. Each of these neurons are again connected to each of the neuron of the output layer and they have randomly initialized weights and biases associated with them. Finally, the linear combination of the inputs, weights and biases are calculated, passed through the activation function and a value is found at the output layer neuron. This value is compared with the label or ground truth value and the cross-entropy loss of the network is calculated. An optimization algorithm can be used to minimize this loss by iteratively updating the weights and biases of the network. All deep neural networks at their core follow this baseline model.

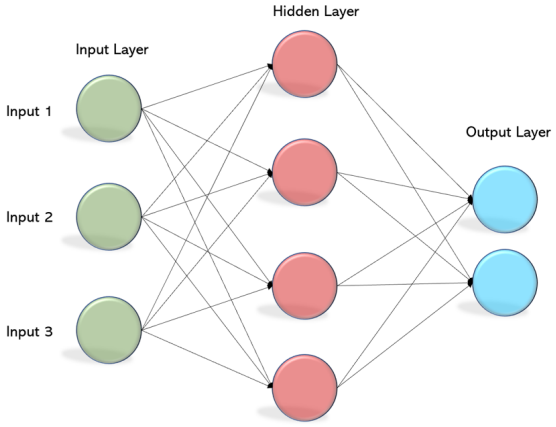

Fig 1. A simple multi-layer perceptron (MLP) architecture. [11]

ii. Convolutional Neural Networks (CNN)

Convolutional neural network or CNN is a subcategory of the FFNN which was inspired by human visual cortex and named after the underlying mathematical operation called a *convolution* operation. CNN is extremely useful for applications where the input feature vector is unusually large and thus the total number of trainable parameters within the neural network architecture goes very high with each addition of a layer. One classic example for such scenario is image classification. Though, an image classification algorithm can be developed using a standard FFNN, it proves to be computationally inefficient for images with higher pixel density.

For example, if an input image is 1000 × 1000 pixels dense and it has three color channels (red, green and blue), then the input feature dimension would be 1000 × 1000 × 3 which is very large and yet only for the input layer. There will also be at least one (usually there are multiple) hidden layers accompanied by their own weights and biases. Thus, for training such a model with hundreds or thousands of images will slow down the computation to an unacceptable extent. This is where CNN can be extremely helpful which can reduce the dimension of the input image going through each layer with the help of the *convolution* operation with appropriate number and dimension of filters/kernels. Few other important components of CNN are the *padding*, *pooling* and *striding* operations which further reduces the feature matrix dimensions and makes the training more efficient.

A popular variation of the CNN is the residual neural network or *ResNet* in short which is especially helpful for cases when there is vanishing gradient issue. Vanishing gradient occurs when the network is too deep (has too many hidden layers) and during back propagation the gradients of the weights and biases sometimes tend to be extremely small which slows down the optimization process. For such cases *ResNet* has been proven to be very helpful which carries a fraction of the linear function from the previous layer's neuron and adds it to the current layer's neuron before passing it to the activation function and thus can avoid the vanishing gradient issue. Due to its benefits, CNN is widely used in computer vision, especially for image classification and object detection which lie at the core of autonomous driving.

iii. Recurrent Neural Networks (RNN)

If several FFNN are stacked on top of each other and the output of one FFNN is passed as the input of the next FFNN, a recurrent neural network or RNN can be constructed. A basic structure of RNN also includes an input layer, one or multiple hidden layer(s) and an output layer. RNN are used for building sequence models which have time dependencies, and the order of the input feature matters. The input vectors are fed into the network in sequence and one vector at a time. With the supplied inputs, after performing some operations such as updating the network weights and biases, the next input sequence is fed to the network.

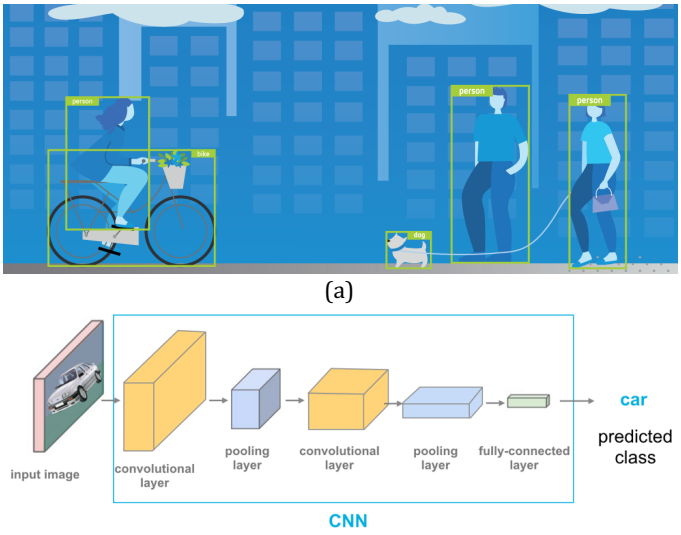

Fig 2. (a) An example of object detection. [12] (b) Basic building blocks of a CNN. [13]

Hidden layers in a RNN architecture can carry the information from the previous layers to the next layers and thus works as a memory. To demonstrate this, the following sentence can be considered- "The instructor, who also possessed an interest in medieval literature, was eager to listen to the student's interpretation of *Dante's Inferno*." In this sentence, the subject-verb agreement between "*The instructor*" and "*was*" occurred after eight words and thus has a long-range dependency. Such dependencies can be tackled by building a sequence model using the RNN. Another popular implementation of the RNN is the long short-term memory (LSTM) network which can deal with even longer dependencies and has been proven to be helpful in multiple scenarios such as time series data, music data and text mining applications.

iv. Generative Adversarial Networks (GAN)

Generative adversarial network or GAN is a combination of a *generator* and a *discriminator* network. The basic concept of GAN is that the generator network aims at creating a fake image from noise and the discriminator network tries to identify whether the generated image is fake or real i.e., if the generated image came from the real training data (the data that was used for building the model). The process is iterative where the main goal of the generator is to convince the discriminator that the images created by the generator are real. The process is said to have converged when the image from the generator and the image from the discriminator are almost indistinguishable from each other.

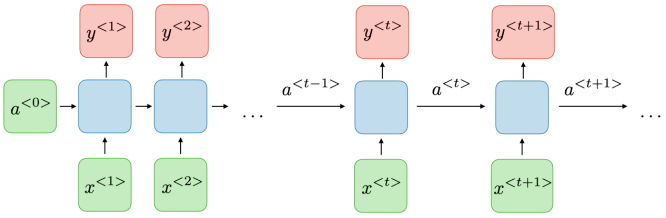

Fig 3. Building blocks of a basic recurrent neural network (RNN) where x is the sequence of inputs at time step t, a is hidden layer activation, and y is the output. [14]

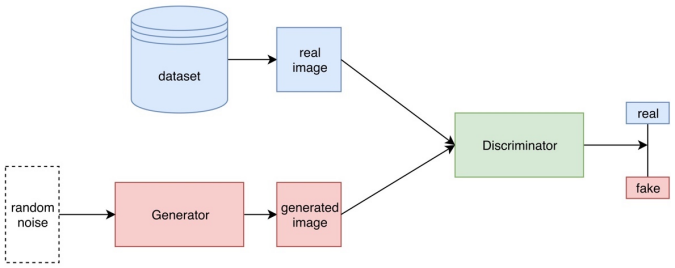

Fig 4. Generative adversarial network (GAN) layout. [15]

A major advantage of the GAN that is worth mentioning is that, once the training phase ends, the discriminator network can be discarded, and the trained generative model is sufficient to work with. Thus, the model complexity reduces to a great extent once the training is complete. Over the last few years, various implementations of the GAN have been found in literature such as Sim GAN, info GAN, Wasserstein GAN and DC GAN. Using the GAN, completely artificial images can be generated that seem to be 100% authentic when compared to the real human being. Also, images of human faces can be created with perfection even though those faces do not exist in real. In the field of text mining, GAN can also be used for text generation.

v. Autoencoders (AE)

Autoencoder (AE) falls under the category of unsupervised machine learning algorithms and thus has no requirement for labelled dataset. AE can sometimes work as an alternative for the principal component analysis (PCA) method for dimensionality reduction. The core idea of an AE is to learn the encoded representation of its input. AE usually consists of two parts-one is the *encoder*, and the other is the *decoder*. The encoder follows a similar approach to the FFNN, and a vector representation of the input is created. The decoder performs something similar to the encoder but in a reverse manner. The AE aims at recreating the input and along the way perform expected improvements on the input such as compression of an input image, reducing noise and even color coding a black and white image. But due its nature, AE can be labelled as a lossy neural network architecture since the output of the decoder is usually only an approximation of the input.

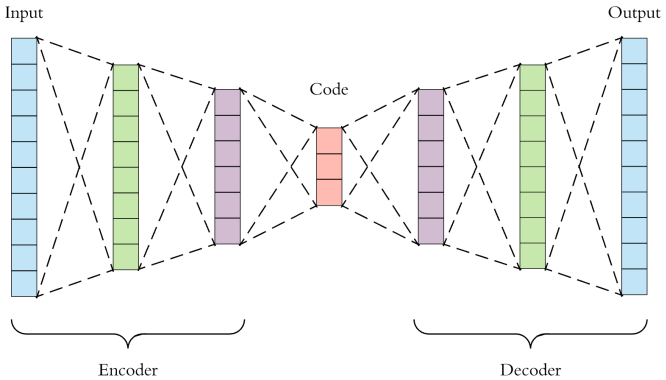

Fig. 5. Autoencoder architecture. [16]

B. Natural Language Processing (NLP)

The field of natural language processing is often referred to as computational linguistics since it aims at working with numerical models to solve practical problems to better understand the human language. The main field of work within the NLP regime consists of language modeling, text morphology and syntactic processing, semantic processing and parsing etc. These aforementioned fields can be considered as the core of NLP; however, it goes way beyond. There are other applications of NLP such as extracting useful information, text and language translations, summary of texts or passage, automatic inference of answers for questions and document classification and clustering as well. The ideas from the core issues are more often than not applicable for solving a practical problem and vice versa.

III. Motivations For DL in NLP

Deep learning is widely used now a days to build forecasting models, fraud detection, computer vision applications such as image classification, object detection and autonomous driving and what not. But what makes the application of deep learning for natural language processing exciting is the intricacy of the in-depth representation of a language using statistical models. Primarily, the deep learning model is supposed to learn the feature representation of the texts or documents to extract meaningful information. Based on these information, further analysis can be performed on the raw data. Despite its complexity, the deep learning methods are worth the extra work because they can outperform the traditional, time-consuming methods that hand-craft the features. Thus, the deep learning algorithms offer a more generalized, data driven approach compared to the trivial methods.

For example, some of the machine learning algorithms previously used were the k-nearest neighbors, naïve Bayes, hidden Markov models, random forests, decision trees and support vector machines etc. While these models perform well for some applications such as article classification and spam emails classification, they do not scale well for dataset where the sequence of the words matter. To put this into perspective, the classic spam classification problem can be considered for which the traditional approach would be using a naïve Bayes classifier on the labeled dataset. However, for this problem, the ordering of the words doesn't matter and working with the word corpus is enough. But for cases such as machine translation, where the sequence of the words is very crucial, none of the classic machine learning algorithm can be useful and rather a deep neural network such as RNN will be more fruitful. While, this is just one example, there are numerous other applications of NLP that require a more robust model for dealing with the text data and thus deep learning is increasingly being used for tackling NLP problems.

IV. DL For NLP Applications

Numerous applications of deep learning can be found in literature for all kinds of practical problems in real life. Rare occasions where natural language processing was dealt with a conventional machine learning algorithm can be found in literature as well. However, in this section of the article, the focus will be on those deep learning implementations that prove to be useful for solving a natural language problem. These NLP applications can be categorized and described as below.

A. Basic Tasks

i. POS Tagging

Part-of-Speech or POS tagging is one of the basic tasks in natural language processing which is the process of labeling words with their part of speech categories. It is also useful for other important tasks such as named entity recognition. Although conventional methods perform quite well for this task, neural network-based

methods have been proposed as well. [17] For example, a deep neural net called *CharWNN* has been developed to combine word-level and character-level representations using CNN for POS tagging. This network emphasizes the importance of character-level feature extraction for producing better results. In another work [18], a variety of neural net models have been proposed for POS tagging such as LSTM, bidirectional LSTM and LSTM with a CRF8 layer etc.

ii. Parsing

Parsing is the process of determining the syntactic structure of a text by analyzing its constituent words. Parsing is performed based on an underlying grammar of the language. Parsing can be further divided into *Constituency* parsing and *Dependency* parsing where the first category refers to the assignment of a syntactic structure to a sentence and the second category refers to the relationship between the words in a targeted sentence. Deep neural nets have been proven to be more useful lately in either of these two types of parsing. For example, a greedy parser that uses vector representation to perform a semantic and syntactic summary of the content has been introduced for Constituency parsing [19]. Also, bidirectional LSTMs have been used for Dependency parsing for feature representation. [20]

iii. Semantic Role Labeling

Semantic role labeling (SRL) is the process of identification and classification of text arguments. It aims at characterizing the sentence elements to determine who did what to whom as well as how, where, and when. The goal of SRL is to extract the semantic relations between the predicate and the related arguments. The predicate refers to what while the arguments consist of the associated participants and properties in the text. Most recent state-of-the-art neural net models employ joint prediction of predicates and arguments, novel word representation approaches, and self-attention models. [21]

B. Text Classification

The primary goal of text classification is to assign predefined categories to text parts (such as a word, a sentence, or an entire document). Preliminarily, it can be used for classification purposes and can be further extended for organization and analysis. A simple example of text classification is the categorization of given documents as to national or international news articles. This NLP procedure can be done with classical ML algorithm such as naïve Bayes or KNN but just like in case of any other NLP applications, DL can offer better results here as well. Both CNN and RNN models have been proposed for text classification. For example, a Dynamic Convolutional Neural Network (DCNN) architecture which is essentially a CNN with a dynamic k-max pooling method was applied to capture the semantic modeling of sentences. [22] An LSTM-RNN architecture has been utilized for sentence embedding in a web search task with high accuracy. However, there are also neural net models that can combine the RNN and CNN for text classification. [23]

C. Text Generation

Many NLP tasks require human-like language generation. For example, article summarization and machine translation convert one text to another in a sequence-to-sequence fashion. Other tasks, such as image and video captioning convert non-textual data to text. Some tasks, however, produce text without any input data (or with only small amount of data). These tasks can include poetry, anecdote and story generation etc. RNN models are usually used for such tasks and they are very efficient at learning the underlying language model. However, these models cannot always produce a structured output or adhere to specific style of text generation. To solve this issue, rather advanced models have been proposed in literature.

Tucker and Kalita [24] generated poems in several languages such as English, Spanish, Ukrainian, Hindi, Bengali, and Assamese where the model was trained on 774 M parameters. Ren and Yang [25] used a basic LSTM model to generate jokes training the model on two datasets. One of the datasets was a collection of short jokes from Conan O'Brien and the other was a set of news articles to generate puns related to current events. Another recent study of interest by Peng et al. [26], used LSTMs to generate stories where he specified whether the story should have a happy or sad ending. Despite the capabilities of DL models to generate human-like written text, there are still issues with the lack of creativity and coherence in the produced texts. However, ongoing research in this field is coming up with more advanced ways for alleviating these issues.

D. Information Extraction

i. Named Entity Recognition

This subtask of information extraction aims at locating and categorizing named entities in context into predefined categories such as the names of people and places. The application of deep neural networks for this task has been done by employing CNN [27] and RNN architectures [28], as well as hybrid bidirectional LSTM and CNN architectures [29].

ii. Event Extraction

A specific type of extracted information from text is an event. Such extraction may involve recognizing trigger words related to an event and assigning labels to entity mentions that represent event triggers. CNNs have been utilized for event detection which can handle problems with feature-based approaches including exhaustive feature engineering and error propagation phenomena for feature generation [30]. In 2018, Nguyen et. Al applied graph-CNN (GCCN) where the convolutional operations are applied to syntactically dependent words as well as consecutive words [31].

iii. Relationship Extraction

This subtask aims at finding the semantic relationships between entity pairs. The RNN model has been proposed for semantic relationship classification by learning compositional vector representations [32]. Also, for relation classification, CNN architectures have been employed as well, by extracting lexical and sentence level features [33].

E. Sentiment Analysis

The primary goal in sentiment analysis is the extraction of subjective information by text mining. It is sometimes called opinion mining, as its primary goal is to analyze human opinion, sentiments, and even emotions regarding products, problems, and varied subjects. Lately, this topic has gained much attention due to its significance in a wide variety of applications and the availability of abundant data. Sentiment analysis can be further divided into sentence level and document level analysis.

i. Sentence Level

At the sentence level, sentiment analysis determines the positivity, negativity, or neutrality regarding an opinion expressed in a sentence. One general assumption for sentence level sentiment classification is the existence of only one opinion from a single opinion holder in an expressed sentence. Recursive autoencoders have been employed for sentence level sentiment analysis by learning the vector space representations for phrases [34]. Long Short-Term Memory (LSTM) recurrent models have also been utilized for tweet sentiment prediction [35].

ii. Document Level

At the document level, the task is to determine whether the whole document reflects a positive or negative sentiment about exactly

one entity. This differs from opinion mining regarding multiple entries. The Gated Recurrent Unit (GRU) neural network architecture has been utilized successfully for effectively encoding the sentences' relations in the semantic structure of the document [36]. Domain adaptation has been investigated as well, to deploy the trained model on unseen new sources [37].

F. Document Summarization

Document summarization refers to a set of problems involving generation of summary sentences given one or multiple documents as input. Generally, text summarization fits into the two following categories.

i. Extractive

Here, the goal is to identify the most striking sentences in the document and return them as the summary. It focuses on sentence extraction, simplification, reordering, and concatenation to convey the important information in documents using text taken directly from the documents. As one of the earliest works on using neural networks for extractive summarization, [38] proposed a framework that used a ranking technique to extract the most salient sentences in the input. This model was improved by [39] which used a document level encoder to represent sentences, and a classifier to rank these sentences.

ii. Abstractive

Here, the goal is to generate summary sentences from scratch; they may contain novel words that do not appear in the original document. Abstractive summaries rely on expressing documents' contents through generation-style abstraction. Since simple attention models perform worse than extractive models, therefore more effective attention models such as graph-based attention [40] and transformers [41] have been proposed in literature for this task.

G. Machine Translation

i. Traditional Approach

One of the first demonstrations of machine translation happened in 1954 [42] in which the authors tried to translate from Russian to English. This translation system was based on six simple rules but had a very limited vocabulary. It was not until the 1990s that successful statistical implementations of machine translation emerged as more bilingual bodies became available [43]. In [44] the BLEU score was introduced as a new evaluation metric, allowing more rapid improvement than when the only approach involved using human labor for evaluation.

ii. Neural Network Approach

The first attempt at neural machine translation (NMT) was taken by Schwenk [45] a feed-forward network was used with seven-word inputs and outputs, padding and trimming. The ability to translate from a sentence of one length to a sentence of another length came into play with the introduction of encoder-decoder models. Such models quickly evolved and were further studied by Cho et al. [46] and numerous novel and effective advances to this model have since been made [47].

H. Question Answering

Similar to summarization and information extraction (IE), question answering (QA) gathers relevant words, phrases, or sentences from a document. In IE a desired set of information has to be retrieved from a set of documents. The desired information could be a specific document, text, image, etc. On the other hand, in QA specific answers are sought, typically ones that can be inferred from available documents.

i. Auditory QA

Smartphones (Siri, Ok Google, Alexa, etc.) and virtual personal assistants are common examples of QA systems with which many interact on a daily basis. While earlier such systems employed rule-based methods, today their core algorithm is based on deep learning. One of the first machine learning based papers that reported results on QA for a reading comprehension test was [48]. Their main contribution was proposing a feature vector representation framework which is aimed to provide information for learning the model. However, more advanced methods have been found in literature. Such as, in [49] CNN was used in order to encode question-answer sentence pairs in the form of fixed length vectors regardless of the length of the input sentence. Also, inspired by neuroscience, incorporated episodic memory in their Dynamic Memory Network (DMN). By processing input sequences and questions, DMN forms episodic memories to answer relevant questions.

ii. Visual QA

Given an input image, visual question answering (VQA) tries to answer a natural language question about the image [50]. VQA addresses multiple problems such as object detection, image segmentation, sentiment analysis, etc. Ref [51] introduced the task of VQA by providing a dataset containing over 250K images, 760K questions, and around 10M answers. [52] proposed a neural-based approach to answer the questions regarding the input images.

I. Dialogue Systems

Dialogue Systems are quickly becoming a principal instrument in human-computer interaction, due in part to their promising potential and commercial value. One application is automated customer service, supporting both online and conventional businesses. Customers expect an ever-increasing level of speed, accuracy, and respect while dealing with companies and their services. Due to the high cost of knowledgeable human resources, companies frequently turn to intelligent conversational machines. Dialogue systems are usually task-based or non-task- based.

i. Task Based

The structure of a task-based dialogue system usually consists of several elements. The first one is the Natural Language Understanding (NLU). This component deals with understanding and interpreting user's spoken context by assigning a constituent structure to the spoken words or sentences and captures its syntactic representation and semantic interpretation, to allow the back-end operation/task. The second component is the Dialogue Manager (DM). The representation generated by NLU would be handled by the dialogue manager, which investigates the context and returns a reasonable semantic-related response. The third component is called Natural Language Generation (NLG). This component produces an utterance based on the response provided by the DM component. Recent task-oriented dialogue systems have been designed based on deep reinforcement learning, which provided promising results regarding performance [53], domain adaptation [54], and dialogue generation [55].

ii. Non-Task Based

As opposed to task-based dialogue systems, the goal behind designing and deploying non-task-based dialogue systems is to empower a machine with the ability to have a natural conversation with humans [56]. Typically, chatbots are of one of the following types: retrieval- based methods and generative methods. Retrieval-based models have access to information resources and can provide more concise, fluent, and accurate responses. However, they are limited regarding the variety of responses they can provide due to their dependency on backend data resources. Generative models, on the other hand, have the advantage of being able to produce suitable responses when such responses are not in the corpus. However, as opposed to retrieval-based models,

they are more prone to grammatical and conceptual mistakes arising from their generative models. Despite remarkable advancements in AI and much attention dedicated to dialogue systems, in reality, successful commercial tools, such as Apple's Siri and Amazon's Alexa, still heavily rely on handcrafted features.

V. Future of DL and NLP

Combining the analysis of all the models, a few common characteristics can be inferred. First, both the convolutional and recurrent methods have made contributions in the recent past, however, a few of attention-powered transformer units as encoders and often decoders, have consistently produced superior results across the NLP field. These models are generally pre-trained in both unsupervised and supervised manner. Second, the attention mechanisms seem to provide the best connections between encoders and decoders. Forcing networks to examine different features usually improves results. Finally, though highly engineered networks can optimize results, there is no substitute for high-volume, high-quality data. From this final observation, it seems useful to direct more research effort toward pre-training methodologies, rather than developing highly specialized components.

One final note for future work can be directed toward a wider variety of languages than it is at present. Currently, the vast majority of research in NLP is conducted on the English language, with another sizeable portion using Mandarin Chinese. In translation tasks, English is almost always either the input or output language, with the other end usually being one of a dozen major European or Eastern Asian languages. Many linguistic intricacies may not be expressed in any of the languages used, and therefore are not captured in current NLP software. Collection and validation of data in under-analyzed languages, as well as testing NLP models using such data, will be a tremendous contribution to not only the field of natural language processing, but to human society as a whole.

VI. Conclusion

Nowadays, highly advanced applications of NLP are abundant. These include Google's and Microsoft's machine translators, which translate more or less competently from a language to scores of other languages, as well as a number of devices which process voice commands and respond as well. The emergence of these sophisticated applications acts as evidence to the impressive accomplishments that have been made in this domain over the last sixty or so years. Without a doubt, incredible progress has taken place, particularly in the last several years.

As has been shown, this recent progress has a clear relationship with the remarkable advances in DNN. Considered an old technology just a decade ago, these machine learning approaches have steered towards progress at an unparalleled rate, breaking performance records in miscellaneous fields. In particular, deep neural architectures, have introduced models with higher performance in natural language tasks, in terms of imperfect evaluation metrics.

While the numerous stellar neural network architectures being proposed are highly competitive, the process of identifying an appealing architecture adds just as much complexity to the problem. In addition to high variability in evaluation data, there are numerous metrics used to evaluate performance on each task. Oftentimes, comparing similar models is difficult due to the fact that different metrics are reported for each. Agreement on particular sets of metrics would go a long way toward ensuring clear comparisons in the field.

Deep learning and NLP are two of the most rapidly developing research topics nowadays. Due to this rapid progress, it is hoped that soon, new effective models will supersede the current state-of-the-art approaches.

References

[1] C. D. Manning, C. D. Manning, and H. Schu¨tze, Foundations of statistical natural language processing. MIT Press, 1999.

[2] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in Advances in neural information processing systems, pp. 649–657, 2015.

[3] K. Cho, B. Van Merrie¨nboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.

[4] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in Proceedings of the 25th international conference on Machine learning, pp. 160–167, ACM, 2008.

[5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 1725–1732, 2014.

[6] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 1717–1724, 2014.

[7] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2107–2116, 2017.

[8] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in International Conference on Machine Learning, pp. 1764–1772, 2014.

[9] C. D. Santos and B. Zadrozny, "Learning character-level representations for part-of-speech tagging," in Proceedings of the 31st International Conference on Machine Learning (ICML-14), pp. 1818–1826, 2014.

[10] B. Plank, A. Søgaard, and Y. Goldberg, "Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss," arXiv preprint arXiv:1604.05529, 2016.

[11] https://becominghuman.ai/multi-layer-perceptron-mlp-models-on-real-world-banking-data-f6dd3d7e998f

[12] https://analyticsindiamag.com/5-object-detection-evaluation-metrics-that-data-scientists-should-know/

[13] https://cezannec.github.io/Convolutional_Neural_Networks/

[14] https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

[15] https://medium.com/dida-machine-learning/data-augmentation-with-gans-for-defect-detection-8318fab1a514

[16] https://predictivehacks.com/autoencoders-for-dimensionality-reduction/

[17] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in International Conference on Machine Learning, pp. 1378–1387, 2016.

[18] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," arXiv preprint arXiv:1508.01991, 2015.

[19] J. Legrand and R. Collobert, "Joint RNN-based greedy parsing and word composition," arXiv preprint arXiv:1412.7028, 2014.

[20] E. Kiperwasser and Y. Goldberg, "Simple and accurate dependency parsing using bidirectional LSTM feature representations," arXiv preprint arXiv:1603.04351, 2016.

[21] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," arXiv preprint arXiv:1703.04826, 2017.

[22] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," arXiv preprint arXiv:1404.2188, 2014.

[23] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward, "Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval," IEEE/ACM Transactions

on Audio, Speech and Language Processing (TASLP), vol. 24, no. 4, pp. 694–707, 2016.
[24] S. Tucker and J. Kalita, "Genrating believable poetry in multiple languages using gpt-2," in Technical Report, University of Colorado, Colorado Springs, 2019.
[25] H. Ren and Q. Yang, "Neural joke generation," Final Project Reports of Course CS224n, 2017.
[26] N. Peng, M. Ghazvininejad, J. May, and K. Knight, "Towards controllable story generation," in Proceedings of the First Workshop on Storytelling, 2018, pp. 43–49.
[27] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," Journal of Machine Learning Research, vol. 12, no. Aug., pp. 2493–2537, 2011.
[28] G. Mesnil, X. He, L. Deng, and Y. Bengio, "Investigation of recurrent neural network architectures and learning methods for spoken language understanding.," in Interspeech, pp. 3771–3775, 2013.
[29] J.P. Chiuand, E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," arXiv preprint arXiv:1511.08308, 2015.
[30] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "Event extraction via dynamic multi-pooling 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), vol. 1, pp. 167–176, 2015.
[31] T. H. Nguyen and R. Grishman, "Graph convolutional networks with argument-aware pooling for event detection," in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[32] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, pp. 1201– 1211, Association for Computational Linguistics, 2012.
[33] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 2335–2344, 2014.
[34] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in Proceedings of the conference on empirical methods in natural language processing, pp. 151–161, Association for Computational Linguistics, 2011.
[35] X. Wang, Y. Liu, S. Chengjie, B. Wang, and X. Wang, "Predicting polarities of tweets by composing word embeddings with long short- term memory," 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), vol. 1, pp. 1343–1353, 2015.
[36] D. Tang, B. Qin and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in Proceedings of the 2015 conference on empirical methods in natural language processing, pp. 1422–1432, 2015.
[37] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large- scale sentiment classification: A deep learning approach," in Proceedings of the 28th international conference on machine learning (ICML- 11), pp. 513–520, 2011.
[38] R. Nallapati, F. Zhai, and B. Zhou, "SummaRuNNer: A recurrent neural network-based sequence model for extractive summarization of documents.," in AAAI, pp. 3075–3081, 2017.
[39] S. Narayan, S. B. Cohen, and M. Lapata, "Ranking sentences for extractive summarization with reinforcement learning," in NAACL: HLT, vol. 1, pp. 1747–1759, 2018.
[40] J. Tan, X. Wan, and J. Xiao, "Abstractive document summarization with a graph-based attentional neural model," in ACL, vol. 1, pp. 1171–1181, 2017.
[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems, pp. 5998–6008, 2017.
[42] L. E. Dostert, "The Georgetown-IBM experiment," 1955). Machine translation of languages. John Wiley & Sons, New York, pp. 124–135, 1955.
[43] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in ACL, vol. 1, pp. 1631–1640, 2016.
[44] H. Schwenk, "Continuous space translation models for phrase-based statistical machine translation," COLING, pp. 1071–1080, 2012.
[45] K. Cho, B. Van Merrie¨nboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," arXiv preprint arXiv: 1406.1078, 2014.
[46] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in NIPS, 2014, pp. 3104–3112.
[47] H. T. Ng, L. H. Teo, and J. L. P. Kwan, "A machine learning approach to answering questions for reading comprehension tests," in Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13, pp. 124–132, Association for Computational Linguistics, 2000.
[48] X. Qiu and X. Huang, "Convolutional neural tensor network architecture for community-based question answering.," in IJCAI, pp. 1305– 1311, 2015.
[49] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in International Conference on Machine Learning, pp. 1378–1387, 2016.
[50] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zit- nick, and D. Parikh, "VQA: Visual question answering," in Proceedings of the IEEE international conference on computer vision, pp. 2425– 2433, 2015.
[52] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A neural-based approach to answering questions about images," in Proceedings of the IEEE international conference on computer vision, pp. 1–9, 2015.
[53] C. Toxtli, J. Cranshaw, et al., "Understanding chatbot-mediated task management," in Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, p. 58, ACM, 2018.
[54] V. Ilievski, C. Musat, A. Hossmann, and M. Baeriswyl, "Goal-oriented chatbot dialog management bootstrapping with transfer learning," arXiv preprint arXiv: 1802.00500, 2018.
[55] J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao, "Deep reinforcement learning for dialogue generation," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1192–1202, 2016.
[56] A. Ritter, C. Cherry, and W. B. Dolan, "Data-driven response generation in social media," in Proceedings of the conference on empirical methods in natural language processing, pp. 583–593, Association for Computational Linguistics, 2011.