A Survey on Language Models

Mohiuddin Qudar
Lakehead University Thunder Bay Campus
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Vijay Mago
Lakehead University Thunder Bay Campus
96 PUBLICATIONS **528** CITATIONS
SEE PROFILE

A Survey on Language Models

MOHIUDDIN MD ABDUL QUDAR and VIJAY MAGO, Lakehead University, Canada

Language models play a fundamentally important role in everyday applications such as grammatical error correction, speech recognition and even in the domain of text summarization. With the recent developments of deep learning methods, traditional *n-grams* and word embedding language models are being replaced with neural network models. As a result, the field of natural language processing has been expanding thus leading researchers to analyze, evaluate and propose new models to solve everyday problems. This paper extensively surveys the recent advancement in language models highlighting their architecture, advantages and limitations, and the datasets used to pre-train or fine-tune the neural language models.

Additional Key Words and Phrases: Natural Language Processing, Language models, Word embeddings, Neural networks

ACM Reference Format:
Mohiuddin Md Abdul Qudar and Vijay Mago. 2020. A Survey on Language Models. 1, 1 (September 2020), 31 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Natural language processing (NLP) is considered to be one of the most important fields of research due to the recent advancements in deep learning models. NLP helps a machine to grasp a high understanding of human language. NLP plays an important role in applications such as understanding textual context, machine translation [1], grammatical error correction [2], speech recognition [3], information retrieval [4], summarization [5], Question Answering [6] and sentiment analysis [7].

Natural language involves a vast number of words or terms which can introduce a wide range of ambiguities [1]. Some forms of ambiguities are lexical ambiguity, where words have multiple meanings [8], syntactic ambiguity [9], where sentence having multiple branches [10], and anaphoric ambiguity [11], where a phrase that refers to a word previously used, but can have numerous outcomes in future. Despite these ambiguities, natural languages are fully understood by humans. However, machines are unable to process the ambiguities of natural human language; therefore, language models are used to translate text to a form that is readable by a machine. They are mainly employed to calculate the probability of a sequence of tokens occurring in a corpus, which is a collection of texts such as a document [4]. In other words, a probabilistic model is developed which helps to predict the next word from a given sequence of words. These fragments or sequence of words are referred to as tokens. Lets consider a partial sentence *Please submit your*. It is more likely that the next word would be *home work* or *paper* than the next word being *Professor*. Also, higher probability for a given sequence of words shows that the sequence of words is more commonly used in that text corpus [12].

Authors' address: Mohiuddin Md Abdul Qudar, mabdulq@lakeheadu.ca; Vijay Mago, vmago@lakeheadu.ca, Lakehead University, ThunderBay, Ontario, Canada.

XXXX-XXXX/2020/9-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Language models are mainly based on Markov assumption, which claims that the *distribution of a word depends on some fixed number of words that immediately comes before it*[13]. This statement is not entirely true, however, it is simpler to use this assumption as the probability distribution of word sequences is not calculated by its arbitrary length but by only fixed number of words that comes before a given word[14].

One of the most common language models are the *n-gram* language models, where *n* is the number of previous words used for calculating a probability distribution [12]. The *n-gram* models generate conditional probability tables by calculating the number of times a word appear in the training corpus [13]. For example the probability of the word *you* after the word *thank*, which can be written as *P(you|thank)* is a conditional probability that can be calculated by the number of times *Thank you* occurs in the corpus divided by the number of times *thank* occurs in the corpus. However, *n-gram* models face difficulties when processing large volumes of short text data such as microblogs, question and answer forums [15], review sites [16], short message service (SMS) [17], email and chat messages. This is because, in platforms such as the SMS, users tend to have various ways of expressing or describing their experiences [17]. In addition, the users are creative at generating texts that in most cases do not follow any grammatical and spelling rules [18] and tend to use more irregular language [16]. Hence, the data has to be parsed keeping in account the context [19].

Recently, with the advancements of deep neural networks, a more flexible approach is proposed [20] for extracting the properties of each word [21]. Then representing the words by using a vector value, as a result the words which are used in similar contexts will tend to have similar vector values [22]. The probability of the next word can then be calculated by using the vector value of the previous words [23].

In spite of a high volume of related works available on language models and on their developments and applications, no comprehensive survey compiling the extensive work done on language models exists so far. This paper is about the recent developments and advances in language models. The following section highlights the research methodology used to select the survey papers, different types of language models, approaches taken for pre-training and fine-tuning the language models, and the datasets used in NLP tasks.

2 METHODOLOGY

This section discusses about the structure and methods used to carry out the survey along with additional information such as current active authors who have made a significant contribution in the recent developments in language modeling.

2.1 Structure of the Survey

Fig. 1 shows the overall structure of the survey. Section 3 discusses the classification of word embedding. Sections 4 and 5 discuss about the types of word embeddings. Section 6 is about the approaches to pretain and fine-tune language models. Section 7 gives a detailed description of the widely used datasets to evaluate or pre-train a language model.

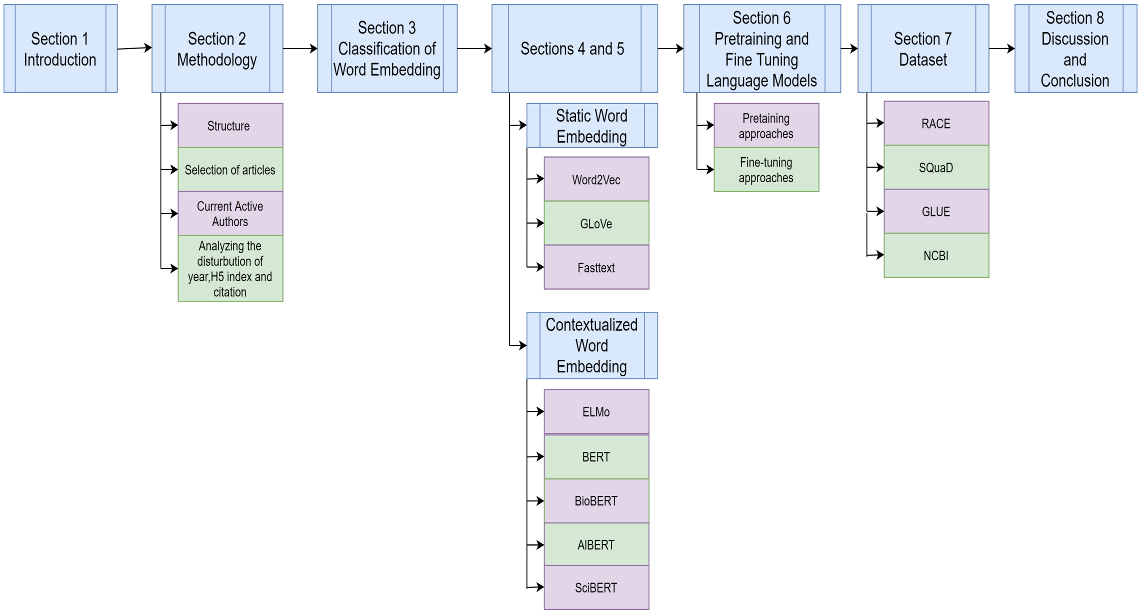

Fig. 1. The structure of the survey

2.2 Selection of Articles

The articles surveyed in this study were selected searching with keywords such as NLP, text classification, sentiment analysis, additionally taking into account the number of citations of the paper and h-index of the venue where the paper was published. Most articles considered are from 2015 onwards. Some of the selected articles are from arXiv, the reason for selecting these articles for conducting the survey is because of the high number of citations the paper has within a short period of time. This reflects the strong impact of the paper on the topic. Table 5 in the Appendix shows the articles selected for the survey, including the total number of citations, h-index on Google scholar of the venue, and the year of publication.

2.3 Current Active Authors in the Field of NLP

For this survey, the name of the authors from the 106 articles that had high citations and high h-index were collected to form a dataset. A word cloud was created to illustrate a visualization of authors who have made a significant contribution in neural language models. As the articles in the survey are from 2015 onwards, the word cloud represents authors currently working in the field of language models. The dataset of authors was pre-processed by attaching the author's first and the last name together. This prevents the repetition of the same author having different names, in the word cloud. Fig. 2 shows the name of the authors. The size of the author's name corresponds to the frequency of that name appearing in the dataset, thus representing a higher contribution of that particular author.

Fig. 2. Names of the authors in the form of word cloud

2.4 Analyzing the Distribution of the Year of Publication and the Citation Range over the Selected Articles

With the recent advancements in deep learning models the importance of NLP has significantly increased and thus a vast amount of research has been conducted. To study the impact of the researches done the total number of citations of the selected articles, h-index of the venue where the articles were published and their year of publications were extensively analyzed. From Fig. 3 it can be seen that no articles were selected from venues that had h-Google index lower than 25 to ensure high quality of articles. In Fig. 4 shows the number of citations each articles has as of April'20. It can be observed that instead of a linear drop there is a increase in number of articles over 3000+ citations. One reason might be most the papers in this citation range are written by authors who made a high contribution, such as Tom Mikolov and Christopher Manning. Fig. 5 gives a visualization of the year of publication of the articles selected for carrying out this survey. It shows most the articles were published recently.

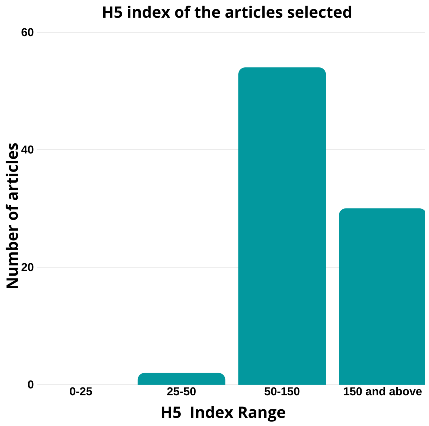

Fig. 3. The h-Google index of the venues from where the articles were selected.

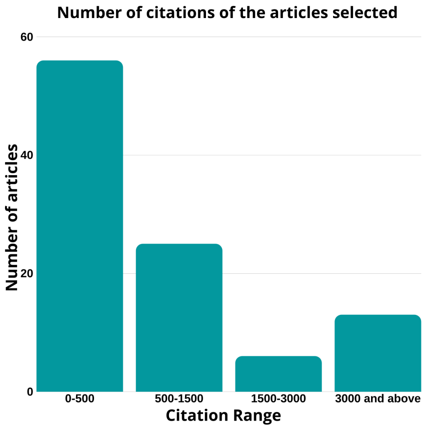

Fig. 4. The total number of citations each articles has as of April'20.

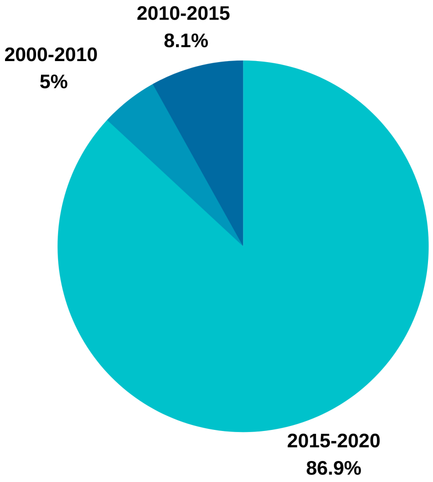

Fig. 5. The year of publications of the articles that were selected.

3 CLASSIFICATION OF WORD EMBEDDING

An embedding is a type of representation for a paragraph or a document where words that have similar meaning have a similar representation [24]. An embedding is a set of methods, in a vector space, where a single word is represented as a real-valued vector [25]. Individual words are converted into a vector value and vector value is represented in such a way it can be mapped into a neural network [26]. As a result, the embeddings are often used with deep learning methods [27]. There are two main types of embeddings static word embeddings and contextualized word embeddings [28]. Static word embeddings consist of word representation methods such as Word2vec, Glove and Fasttext [29] whereas from contextualized word embeddings recent language models are developed such as ELMo and BERT, which are introduced later in section 4. Fig. 6 gives an overview of how the current neural language models are developed from embeddings. This survey focuses on the recently developed language models such as ELMo and BERT.

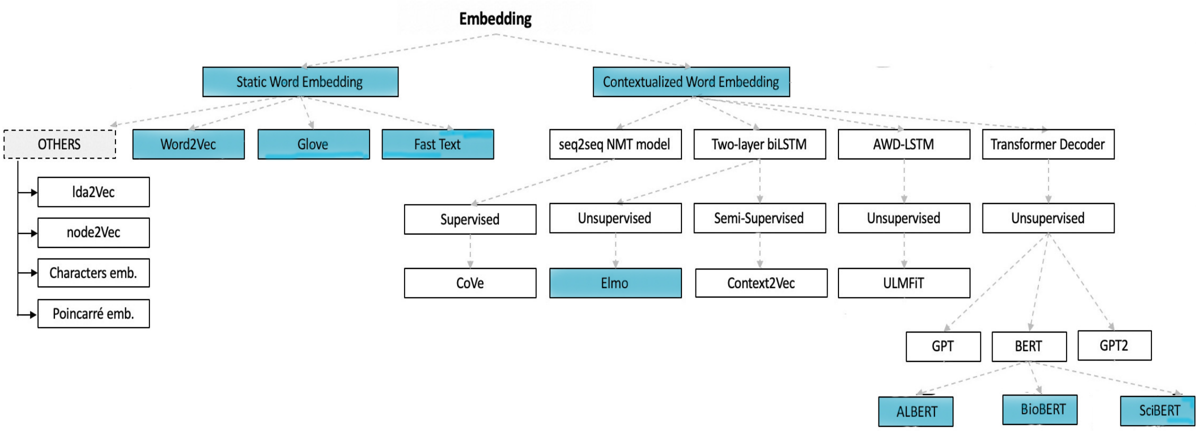

Fig. 6. An overview of evolution of embeddings. Focus of this survey is highlighted in blue color.

4 STATIC WORD EMBEDDING

Word embedding is a form of word representation, where words are represented using real-valued vectors. These vectors represent information about a language, like word semantics [9]. Word embedding is also called semantic vector space [30] or semantic model [31] because it groups and classifies similar words together [32]. For example fruits like orange, grape, apples will be grouped together whereas cars would be far away from that group. In other words, word representation constructs a vector representation of fruits that is different from the vector representation of cars. Word embeddings are used for various applications such as computing similar words [24], document clustering [33] or sentimental analysis [34]. Word embeddings are also helpful when extracting features for text classification because text-based classifier models cannot be trained on strings [35], since machines can only compute vectors of numbers [36]. As a result, the text is converted into an array of vectors which is then fed to the model for training. Table 1 shows the differences between the two types of word embeddings.

4.1 Word2Vec

For constructing word embedding a method called word2vec is used. Word2vec tries to capture the linguistic context, syntactic and the semantic word relationship by clustering similar meaning words together and placing dissimilar words "far away" [37] [40]. It is a two-layered neural network [42], where there is only one hidden layer and an output layer. It is not a deep neural network but it converts text to vectors which is recognized by deep neural networks [38]. The output of word2vec is a vocabulary where each word has a vector value attached to it. Table 2 shows an example consisting of a list of words, with their associated cosine distance, in relation to the word "Sweden".

Parameters	Static Word Embeddings	Contextualized Word Embeddings
SEMANTICS	Does not consider the polysemy of a word [24], which is the the ability of words to have multiple meaning. Same embeddings are generated from same word in different contexts [37].	Considers the word semantics in different context taking into account the context of a word [38].
OUTPUT	The output of the training models,for example word2vec, are only vectors [39].	The output of the training is a trained model and vectors [38]. As a result, this trained model can be used to fine-tune different NLP tasks, such as SQuad [6].
NLP TASKS	Word vectors have very shallow word representations [40]. In other words, it only has a single layer for training and each time the network has to be trained from scratch to fine-tune on a NLP task [37].	Weights from the trained model generated can be used to fine-tune the models for a specific natural language task [33] [28]. This process is called transfer learning where instead of training a model from scratch existing neural network models can be modified to train on a small data and give high performance [41].

Table 1. Differences between static and contextual word embeddings

Word	Cosine Similarity
Norway	0.76
Denmark	0.71
Finland	0.62
Switzerland	0.58
Belgium	0.57

Table 2. List of word associated with the word Sweden in order of similarity [30]

Cosine similarity is used to find the similarity between two vectors. Value one represents similar vectors or words whereas zero represents no similarity between vectors [30]. For example, to find the similarity between the sentences *Have a good day* and *Have a great day*, a list of words from both texts are made *Have, a, good, great, day* then the number of times each of these word appears in each text is counted and shown in Table 3

$$\cos\theta = \frac{A.B}{||A|| \times ||B||} \tag{1}$$

	have	a	good	great	day
Sentence A	1	1	1	0	1
Sentence B	1	1	0	1	1

Table 3. Number of times each of the word appears in a text

Cosine similarity is calculated to be 0.8 using Equation 1 where A is the vector [1,1,1,0,1] and B is the vector [1,1,0,1,1]. This shows that both the sentences have similar linguistic context.

4.2 Global Vectors for Word Representation(GLoVe)

GLoVe was introduced to address one of the main disadvantage of word2vec. Word2vec only relies on *local context information of language* [43]. In other words, the semantics captured to form word representations only uses information from the surrounding word [44]. GLoVe, on the other hand, captures both the local statistics and global statistics to form word representations [43]. It creates word vectors that takes into account the meaning of a word in respect to its context [45]. GLoVe learns based on global word to word co-occurrence matrix and train word vectors to derive semantic relationships between words from a constructed co-occurrence matrix [46]. For example, GLoVe takes into account the local context for the sentence: *The cat sat on the mat* by computing a co-occurrence matrix.

	the	cat	sat	on	mat
the	2	0	0	0	0
cat	0	1	0	0	0
sat	0	0	1	0	0
on	0	0	0	1	0
mat	0	0	0	0	1

Table 4. Shows word co-occurrence matrix on a given sentence [28]

The matrix shown in Table 4 represents the number of times each word appear in a sentence. GLoVe can be used to find relationships between similar words such as synonyms or zip codes and cites [43].

4.3 FastText

Fasttext is a word embedding model where a word vector is built based on *n-gram* models and each word is represented as a bag of character *n-grams* [47][48]. As a result, the overall word embedding is a sum of the character *n-grams* [27]. For example, if $n = 3$, and the vector word is *language* then the fasttext representations for the character *n-grams* is $< la, lan, ang, ngu, gua, uag, age, ge >$. The symbols $< and >$ denotes the end and beginning of a word, and prevents associating one word with *n-grams* of other short words. For instance *age* from *language*. Fasttext is useful as words that do not occur in the training corpus are assigned a vector value for its characters [27].

5 CONTEXTUALIZED WORD EMBEDDING

The fundamental task of language modeling is to estimate a probability distribution over a sequence of words [49]. A Language model provides context, to distinguish between words that sound similar like *ice cream* and *I scream* but have different meanings, or to predicts the probability of a certain word occurring in a context [39]. This section discusses about some of the prominent language models such as ELMo, BERT and BioBERT, which have been introduced recently. These language models show significant increase in performance among NLP tasks for example in the field of text classification [18], information retrieval [4] and paraphrasing [4]. Contextualized word embedding can be of two types neural and count-based language model [50].

- Count-based language model or *n-gram* language model is a probabilistic language model for predicting the next word in a given sequence of words [7]. Given a sentence or paragraph s, length n, language model assigns a a probability $P(s) = P(w_1, w_2, \ldots, w_n)$ to the whole sequence [51]. The probability of this word sequence can be further broken down into the product of conditional probability of the next word given the previous words or context. $P(w_1)P(w_2/w_1)...P(w_n/w_1,w_2...w_{n-1})$ where w_n represents the n^{th} word in the sequence [39].
- The underlying problem that makes language modeling difficult is the data sparsity. Data sparsity is the term used to describe the phenomenon of not having enough data in a corpus when a model is trained [52]. This leads to a model predicting a probability of zero for a word that did not occur in the training corpus [39]. As a result, large training corpora are used. However, as the size of the corpora increases the number of features, parameters and dimensions also increases. With higher dimensions sparsity of data again occurs, as the data are not close enough, shown in Fig. 7 [28]. This effect is known as the curse of dimensionality [53]. To reduce the affect of the curse of dimensionality neural network language models has been introduced [54]. Neural networks are instantiated with a certain number of features or dimensions and each aspect of a data can fall under each dimension. For instance, if a fruit is considered to be a data then the features can be color, weight and shape. The features would add these information and predict the fruit that is being considered [55].

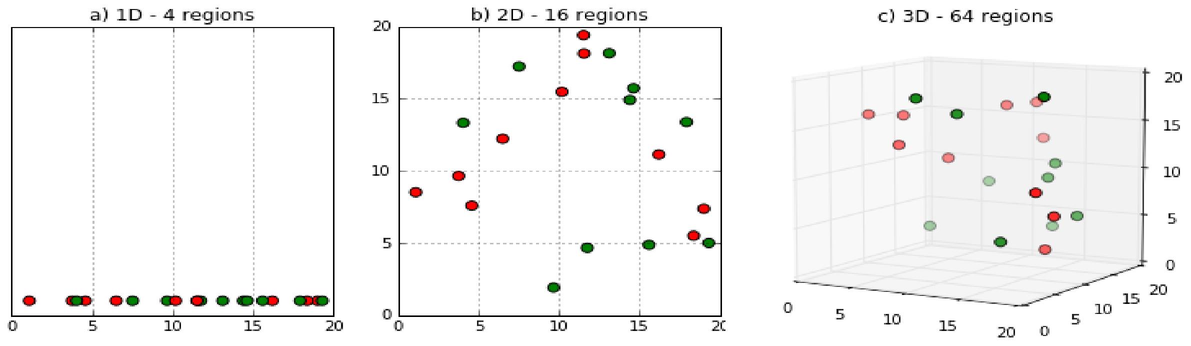

Fig. 7. Shows how the data sparsity increases with the increase in dimensions[54]

5.1 Embeddings from language models (ELMo)

One of the fundamental components in most neural language models are pre-trained word representations [56]. As discussed above, word representation is a widespread idea in NLP where words are represented using vectors [14]. Though word representation is an important component, it can be difficult to train high quality representations because in practice it does not take into account the syntactic characteristics and ambiguities [57]. Syntactic characteristics are the complex characteristics of words [8] and ambiguities are how meaning of a word varies across a context [8]. To address both the complex characteristics of words and ambiguities embeddings from language models (ELMo) have been introduced[56].

ELMo is a deep contextualized word representation feature based approach which has achieved outstanding performance over a wide range of natural language problems [56]. The word representations in ELMo are quite different from conventional word representations, where a sequence of words or tokens are assigned a vector value that is a subset of the entire input sentence[56]. In ELMo, a bidirectional language model is used to pre-train a large text corpus[56]. A bidirectional language model helps to look at a word from both left to right and right to left contexts to see how that word fits in a given sentence[56]. ELMo has three main features for word representations. They are:

- Contextual, because the representation of a given word depends completely on the context in which it is used [8].
- Deep, since the representations combines all the layers of the neural network [56].
- Character-Based, as the representations are entirely based on character making the neural network effective against morphological words to form representations for tokens not found while training the corpus [56].

5.2 Bidirectional Encoder Representations from Transformers (BERT)

BERT is similar to ELMo, but uses a pre-trained neural network instead of feature based approach for word representations [18]. BERT's key component is that it applies bidirectional transformer language model while training a corpus [18]. A Transformer is a machine learning model that takes into account the ordered sequence of the data, even though it is not necessary that the sequence of words are processed in that order[46]. As a result, it can start to process the end of a sentence without starting to process the beginning. If a language model is trained using a bidirectional Transformer it can have a sense of the linguistic context [19]. BERT used two training techniques Masked language model and Next Sentence Prediction [18].

5.2.1 Masked Language Model. Although it is logical to think that bidirectional model performs better than unidirectional models but bidirectional model has its own disadvantage. When using a corpus to train a bidirectional model, it can "allow each word to see itself" [18], since it is bidirectional in nature. To solve this, some percentage of words are randomly masked and the model is asked to predict random words from the input rather than the next word from the sequence [46]. Masking is carried out in three different ways. For example if the sentence to be trained is "My dog is hairy" [18] and the word "hairy" is chosen to be the token, then masking is done either by replacing it with a $< Mask >$ token i.e., "My dog is $< Mask >$" or with a random token e.g. "My dog is apple" or keeping it as it is i.e., "My dog is hairy" [18]. Using these three ways together masking is done to capture the contextual meaning of a word. If only the first method was used, that is only using $< Mask >$ tokens, then the performance of the model would be low as it was never trained on anything other than a masked object. Also sometimes keeping the sentence intact, the model is forced to train on the original representation of the sentence to introduce biasness [46]. This biasness helps the language model to stick to the context [58].

5.2.2 Next Sentence Prediction. The second part for pre-training BERT is done by a method called Next Sentence Prediction [59]. This method requires giving the model a pair of sentence and then testing if the model can predict whether the second sentence comes after the first sentence or not in the corpus. 50 % of the time the second sentence is actually related to the first sentence [18]. Next Sentence Prediction is mainly carried out so that the model can understand and relate how two sentences are connected

[46], and this helps the model to perform better in various NLP tasks such as Language Inference [4] or Question Answering [15].

5.3 Bidirectional Encoder Representations from Transformers for Biomedical Text Mining(BioBERT)

Biomedical text analysis is becoming very popular due to the rapid increase in the number of biomedical documents[60]. For instance, around 3000 articles are published in peer-reviewed journals every day [61]. Moreover, on PubMed, a search engine that contains medical literature and biomedical information has more than 30 million citations and abstracts as of January 2020 [61]. As a result, with recent advancements in NLP and language models extracting valuable information from biomedical documents has become widespread [58]. However, directly using natural language model on biomedical corpora gives very poor performance because due to a change in word representation from a general domain [16] to a biomedical domain [62]. Therefore, to address this issue BioBERT was introduced.

BioBERT is a domain specific language model and pre-trained on wide range of biomedical corpora [63]. BioBERT has the same structure as BERT [62]. Like BERT, BioBERT was first pre-trained on general domain such as English Wikipedia [1] and BooksCorpus [18] and then pre-trained explicitly using biomedical corpora such as PubMed Abstracts, PMC Full-text articles to make the language model domain specific [62]. Thus, BioBERT gives a better performance than BERT and other previous language models for biomedical text mining tasks [64], such as biomedical named entity recognition [15], National Center for Biotechnology Information(NCBI) disease corpus, biomedical relation extraction [63] and biomedical question answering [63].

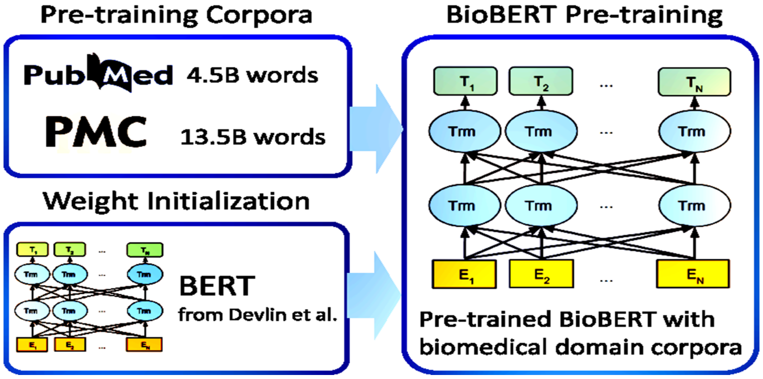

Fig. 8. Pre-training of BioBERT with words from PubMed and PMC [62].

5.4 A lite BERT (AlBERT)

Generally, increasing the number of training corpus and the model size increases the performance of the NLP tasks[65]. However, as the model size increases it becomes difficult to pre-train the model because of the "GPU/TPU memory limitations and longer training times" [65]. To solve this issue a lite BERT(AlBERT) was introduced. AlBERT has the same architecture as BERT, and it also uses Bidirectional Transformer. AlBERT uses two parameter-reduction techniques to significantly reduces the number of training parameters of BERT. They are:

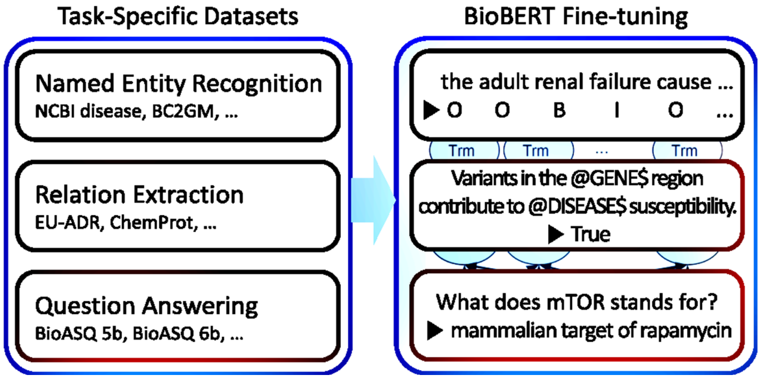
Task-Specific Datasets
Named Entity Recognition
NCBI disease, BC2GM, ...
Relation Extraction
EU-ADR, ChemProt, ...
Question Answering
BioASQ 5b, BioASQ 6b, ...
BioBERT Fine-tuning
the adult renal failure cause ...
▶ O O B I O ...
Trm
Trm
...
Trm
Variants in the @GENE$ region contribute to @DISEASE$ susceptibility.
▶ True
What does mTOR stands for?
▶ mammalian target of rapamycin

Fig. 9. Finetuning of BioBERT on specific NLP tasks [62].

- Factorized embedding parameterization [65], it breaks down the large word matrix into smaller matrices. As a result the size of the word representations is separated from the size of vocabulary embedding [66].
- Cross-layer parameter sharing, which stops the parameters from increasing as the depth of the neural network increases [65].

Both the techniques significantly decrease the training time and increase the training speed of the model [65].

5.5 A Pre-trained Language Model for Scientific Text (SciBERT)

Unsupervised pre-training of language models on large corpora, for example in the case of ELMo and BERT, significantly improves the performance of many NLP tasks [66]. However, ELMo and BERT performs poorly when extracting information from scientific articles because they are pre-trained on general domain corpora such as news articles and English Wikipedia [67]. To extract information from scientific articles SciBert have been introduced. SciBert is also a pre-trained language model based on the achitecture of BERT, but unlike BERT, SciBert is trained on a large corpus of scientific text making it domain specific [68]. As a result, SciBert shows significant increase in performance on a range of NLP tasks [69][70][71] in the scientific domain. Building a large scale training data is often possible through crowd sourcing but for scientific domains getting annotated data is difficult and expensive as it requires the help of experts for quality annotations [66]. The development of SciBert is discussed below.

- *Vocabulary:* For unsupervised tokenization for the input text BERT uses a vocabulary called WordPiece [72], that contains the most frequently used words or subwords used. However, for tokenizing SciBert a new WordPiece was developed called SentencePiece [73]. SentencePiece contains the most frequently words used in scientific domains [74]. A comparitive study was done to find out the overlapping words between SentencePiece and WordPiece and it showed a similarity of only 42% [75]. This demonstrates that there is a significant difference in the frequently words used between scientific and general domain texts [75].
- *Corpus:* To make SciBert domain specific on scientific tasks, it is trained on "random sample" of 1.14M papers from semantic scholar [76]. The corpus from semantic scholar consists of 20% papers from the computer science domain [77] and 80% papers from biomedical domain [78]. For training not only just the abstracts but full text of the papers are used. An average paper has around 2700

tokens [79], as a result the dataset contains more than three billion tokens. The number of tokens used to pre-train SciBert is similar to the number of tokens used to pre-train BERT.

6 PRE-TRAINING AND FINE-TUNING LANGUAGE MODELS

This section discusses the steps and the approaches taken to pretain and fine-tune language models. It also discusses about language models that performed significantly better when pre-trained and fine-tuned in a specific way.

6.1 Pre-training Approaches

Extracting meaningful information from corpus is very important which has been categorized into document level extraction and sentence level extraction [80]. This method can be further be broken down into feature based and neural based. Feature based method mainly focuses on extracting the important features whereas, the neural based methods specialize in learning the features from the neural networks automatically [7]. To generate enriched training data, with important features which include labeling events from unsupervised corpus, multiple event generation external resources are used. Unsupervised corpus are corpora which automatically generates training data [81]. Also, abstract meaning representation and distribution semantics have been used to extract events from corpus and additional events have also been extracted by using frames in FrameNet [51]. Language models have also been pre-trained from supervised translational corpus where ELMo [2] [51] uses sensitive embeddings by encoding characters with stacked bidirectional LSTM (Long Short Term Memory) [82] and residual structure [83]. By using ELMo, effective results were obtained on text classification. The breakthrough occurred when BERT language model was used, which improved performance on 11 NLP tasks including the Stanford's Question Answering dataset(SQuAD) [18] [6] [84]. In addition, researchers have also conducted exhaustive researches in learning the different representation of words efficiently by using both neural and non-neural approaches [85]. pre-trained word embeddings are an important block of modern NLP systems, pre-trained word embeddings have helped to achieve significant text classification performance [86]. In order to pre-train word embedding vectors, left-to-right language modeling objectives have been applied [60] and also other approaches were used which includes distinguishing correct words from incorrect words in left and right context [14]. These methods have also been used in sentence embeddings and also paragraph embeddings [87]. For training sentences, next sentences which might be following a given sentence is ranked by training sentence representations and also generating sentences from left to right from a given representation of the previous sentence[88] [20].

6.2 Fine-tuning Approaches

Recently, unlabeled text has been used to pre-train sentence or documents which constructs contextual token representations and has latter been fine-tuned with labeled text specifically for supervised downstream task that heavily utilizes pre-trained language models [5]. The major benefit of utilizing such approaches is that only a few parameters are needed as prerequisites, in order to learn a corpus from the beginning. Due to this particular benefit, OpenAI GPT, a language model, has significantly improved the sentence level NLP tasks from the the GLUE benchmark [89]. The OpenAI GPT language model has been designed with left-to right architecture in which the tokens of a sentence are evaluated from a left to right direction in which every token can only be attended to the preceding tokens in the self-attention layers of the Transformer [50] [19]. However, for particularly sentence level NLP tasks such approaches has shown to be inefficient this is because when fine tuning based approaches are applied to such token level tasks, such as question answering [46], it is highly important for the language model to process contexts from both the directions, which the OpenAI GPT is unable to do [90]. Also, research has been emphasized on effectual transfer from supervised tasks with the aid of large datasets, such as natural language inference [91] and machine translation [22]. Transfer learning from large pre-trained models has also shown to be highly important in the field of computer vision as well in which pre-trained models have been fine-tuned with ImageNet dataset [92] [93].

7 DATASETS

In this section, datasets used for evaluating different language models are described below in details. The datasets are either used for training a language model or for fine tuning to increase a model's accuracy for a particular natural language task [58].

7.1 Large-scale ReAding Comprehension Dataset From Examinations

Large-scale ReAding Comprehension Dataset From Examinations(RACE) is a collection of approximately 28,000 English passages and 100,000 questions. It was generated by instructors who taught English in middle and high school to students between the age of 12 to 18. It was comprehensively constructed to assess the students' skills in "understanding and reasoning" [94]. As a result, the percentage of questions that needs reasoning is greater compared to other datasets like SQuAD [6] or NCBI [63]. Unlike RACE, existing datasets do not consider the reasoning variable and have two main limitations. They are:

- For a NLP task, for example in question answering dataset, the results can be directly mined from the text with a simple word-based search [59] and comparing the context [16]. Thus the model does not take into account the reasoning and this in fact also restricts the types of questions [95].
- Many NLP datasets are either crowd sourced [6] or computer generated; this introduces various types of noises, for example domain experts [96] only achieved an accuracy of 82% in the dataset, Who-did-What[94], because of the biasness that was introduced due to the way the data was collected [97].

To address the above limitations RACE was developed from English exams to assess a student's reading skills. Fig. 7.1 shows a sample from the dataset. This dataset was designed in a unique way to the evaluate a model's ability in reading passages. Like multiple choice questions [96], the dataset has several questions and each question has a set of four answers, from which only one of them is correct. However, the answers in RACE do not depend on the original text because they can be deduced only through critical thinking. Therefore the questions can be worded using words not present in the training corpus. The main advantages of RACE are discussed below:

- RACE can be used as an indicator for measuring a model's capability to understand a comprehension. This is because the questions and answers are created by humans who are experts in the English language field for evaluating the students' reading ability [94].
- RACE is the largest dataset considering the number of questions that needs logical reasoning to answer. Because of the size of the dataset, it can support the training of neural network language models. For instance, MCTest [96] was developed to assess the reasoning factor in machine language models [94] through crowd sourcing, but consists only of 2,000 questions, which is very low for training a language model.
- The answer options in RACE are human-generated sentences, which means there are options for an answer that might not be seen in the training corpus. This allows a diversity of questions to be asked [59].
- RACE covers a wide range of domains and numerous types of writing. This allows language models to be to be generic, unlike models like BioBERT[62] [94] that are domain specific.

Passage: Do you love holidays but hate gaining weight? You are not alone. Holidays are times for celebrating. Many people are worried about their weight. With proper planning, though, it is possible to keep normal weight during the holidays. The idea is to enjoy the holidays but not to eat too much. You don’t have to turn away from the foods that you enjoy. Here are some tips for preventing weight gain and maintaining physical fitness: Don’t skip meals. Before you leave home, have a small, low-fat meal or snack. This may help to avoid getting too excited before delicious foods. Control the amount of food. Use a small plate that may encourage you to ”load up”. You should be most comfortable eating an amount of food about the size of your fist. Begin with soup and fruit or vegetables. Fill up beforehand on water-based soup and raw fruit or vegetables, or drink a large glass of water before you eat to help you to feel full. Avoid high-fat foods. Dishes that look oily or creamy may have large amount of fat. Choose lean meat . Fill your plate with salad and green vegetables. Use lemon juice instead of creamy food. Stick to physical activity. Don’t let exercise take a break during the holidays. A 20-minute walk helps to burn off extra calories.
Questions: What is the best title of the passage?
Options:
A. How to avoid holiday feasting
B. Do’s and don’ts for keeping slim and fit.
C. How to avoid weight gain over holidays.
D. Wonderful holidays, boring experiences.

Fig. 7.1. A sample from RACE dataset showing there are multiple choice for a given question [94].

7.2 The Stanford Question Answering Dataset

The Stanford Question Answering Dataset (SQuAD) is a collection of more than 100,000 questions answered by crowdworkers on a set of Wikipedia articles in which the answer to every question is a fragment of text from a reading passage. SQuAD contains 107,785 question-answer pairs on 536 articles [6]. In the dataset, each question and its following answer is presented with a passage from Wikipedia containing the answer [6]. The dataset is mainly designed for a machine to predict the answer from the context of the passage, as reading and understanding comprehension is itself a challenging task for machines [96]. A machine needs to not only understand the natural language but also needs to extract the rich knowledge from a comprehension as well. Also, in the field of machine learning, large detailed datasets have played an important role, such as the ImageNet dataset, used for object detection. This dataset has a collection of more than 14 million images manually annotated by humans and trained on machines so that objects can be predicted more efficiently [22]. Fig. 7.2 shows a sample of the SQuAD dataset.

Although some reading comprehension datasets like RACE are high in quality, but these datasets are not enough for training the present-day upgraded language models. Nonetheless, the existing datasets are quite large, but they do not follow the same characteristics as reading comprehension questions tend to follow [6]. Thus, a linguistically high quality and large SQuAD v1.0 was presented. Unlike the previous reading comprehension datasets such as the MCTest dataset [96], the SQuAD dataset does not provide a list of choices for each of the questions. This is because the dataset has been designed for a machine to select the answer from all the possible contexts in the passages. Although the questions are restricted and not as interactive as most of the modern standard tests, a rich variety of question and answers types were traced in their dataset. In order to evaluate the level of difficulty of the SQuAD dataset, logistic regression model was implemented with a set of features, in which they have found that lexicalized and dependency features are extremely important to improve the performance of the model [43]. Dependency features are based on the occurrence of each word in both the question and sentence. On the contrary, the performance of the model decreases when the complexity of the answer types increases and also when syntactically the diversity between the question and the sentence containing the answer increases [6]. However, humans were still able to answer the questions even though the complexity of the answer types and the diversity were increased syntactically[43]. Overall, their model received F1 score of 51.0%. Significant improvements were observed when neural network-based models were used in which they have obtained F1 score of 70.3% on SQuAD v1.1 [18]. However, these results were still far behind compared to the human performance of F1 score of 86.8% [18]. However, the language model BERT has outperformed the human performance with an F1 score of 93.2% [18].

> **Passage:** Apollo ran from 1961 to 1972, and was supported by the two-man Gemini program which ran concurrently with it from 1962 to 1966. Gemini missions developed some of the space travel techniques that were necessary for the success of the Apollo missions. Apollo used Saturn family rockets as launch vehicles. Apollo/Saturn vehicles were also used for an Apollo Applications Program, which consisted of Skylab, a space station that supported three manned missions in 1973–74, and the Apollo–Soyuz Test Project, a joint Earth orbit mission with the Soviet Union in 1975.
>
> **Question:**
> What space station supported three manned missions in 1973-1974
>
> **Answer:**
> Skylab

Fig. 7.2. A sample from SQuAD dataset [6].

7.3 The General Language Understanding Evaluation (GLUE)

The General Language Understanding Evaluation benchmark (GLUE) consists of datasets used for "training, evaluating, and analyzing" language models [89]. GLUE has a collection of nine distinct datasets designed in such a way so that it can evaluate a model's language understanding. GLUE datasets are based on English sentence understanding tasks which covers a wide range of domains [18]. The datasets are either single sentences or sentence-pair language understanding tasks made from existing datasets such as question answering [6], sentiment analysis [89] to cover a broad range of "text genres, and degrees of difficulty". GLUE also has an online platform for evaluating a model's performance and displaying the performance on a public leaderboard. Detailed descriptions of the nine different dataset are given below:

- Corpus of Linguistic Acceptability(CoLA): CoLA is a single-sentence task consisting of more than 10,000 English sentences drawn from various books and articles on "linguistic theory" [98]. Each sentence is annotated with whether the sentence is grammatical or ungrammatical English sentence. Using such a binary single-sentence classification dataset the goal of language model is to predict if a sentence is linguistically correct or wrong.
- The Stanford Sentiment Treebank(SST): SST is also a binary single-sentence classification task containing sentences from movie review and their sentiment labeled by humans [99]. The task of language model is to predict the sentiment of a given sentence only.

- The Microsoft Research Paraphrase Corpus(MRPC) MRPC is a sentence pair corpus generated from online news sources, with human annotations for whether both the sentences are semantically equivalent or not. Thus, the task is to predict if a given sentence-pair has semantic similarity or not [89].
- The Quora Question Pair(QQP): QQP is similar to MRPC, the task is to predict how similar a given pair of questions are in terms of semantic meaning [89]. However, unlike MRPC, QQP dataset is a collection of questions from the question-answering website Quora [1],
- The Semantic Textual Similarity Benchmark(STS-B): STS-B is a collections of sentence pairs extracted from news headlines, video and image captions and similar sources and based on the semantic similarity the sentence pairs are given scores from 1 to 5. The task is to predict the scores [100].
- The Multi-Genre Natural Language Inference Corpus(MNLI): MNLI is a crowd sourced dataset, consisting of sentence pairs with a human annotated premise and a hypothesis sentence. The task is to predict whether the premise sentence "entails" the hypothesis, contradicts the hypothesis sentence or stays neutral[89].
- Question Natural Language Inference(QNLI): QNLI is a simplified version of SQuAD dataset which has been converted into a binary classification task by forming a pair between each question and each sentence in the corresponding context . A language model's task would be to determine if the sentence contains the answer to the question. A positive value is assigned if pairs contain the correct answer, similarly a negative value is assigned if the pairs do not contain the answer [89].
- Recognizing Textual Entailment(RTE): RTE is similar to MNLI, where the language model predicts if a given sentence is similar to the hypothesis, contradicts or stays neutral. RTE dataset is very small compared to MNLI. [99].
- The Winograd Schema Challeng(WNLI): WNLI is a reading comprehension task in which a model takes a sentence with a pronoun as an input and selects an answer from a list of choices that references to the given pronoun [98].

7.4 National Center for Biotechnology Information Disease Corpus (NCBI)

In the domain of biomedical corpus, Named Entity Recognition (NER) plays an extremely vital role in biomedical text mining tasks, which consists of identifying domain specific proper nouns. NER is defined as an NLP task that identifies and classifies named entities from unstructured texts into pre-defined categories such as person names, organizations, locations [101]. In biomedical literature publications, the semantic contextual information can only be useful if efficient and trustworthy tools are readily available for extracting and analyzing such biomedical information. Hence, NLP and text mining tools are highly crucial for collecting important information. However, diseases are detected automatically, by text miming tools, only when domain-specific annotated corpora are easily accessible [9].

The NCBI disease corpus is a collection of 793 PubMed abstracts in which each abstract is manually annotated by two annotators by tagging the name of the disease and their corresponding concepts in Medical Subject Headings [16] or in Online Mendelian Inheritance in Man [102]. In addition, fourteen annotators were randomly paired so that when there were disagreements between the annotators for not agreeing with the same annotation, an agreement could be reached between the annotators by further discussions in two annotation phases. Ultimately, all the results of the annotation were checked against the annotation of the rest of the corpus for reassuring a corpus-wide consistency in the results [11].

[1] https://www.quora.com/

One of the popular biological attributes in the field of biomedical research are diseases, which are repeatedly researched in literature as well as on internet. Similarly, like NER tasks, disease name recognition is also considered as an important task in biomedical text mining; however, identifying diseases from unstructured data is a vastly difficult task due to its variation and significant ambiguity in many of the disease names. For example, *adenomatous polyposis coli* [64] and *Friedrich ataxia* [64] are considered as both gene and disease names. Also, abbreviated disease names are commonly used in biomedical texts, such as AS can stand for *Angelanguage modelan syndrome, ankylosing spondylitis, aortic stenosis, Asperger syndrome* or even *autism spectrum* [15]. Although, doctors and health practitioners have their own ways of describing a disease, it may be even more difficult to implement automatic identification methods into such medical texts as it will be harder for the language models to work more effectively.

Disease name entity recognition methods consist of two separate steps:

- Disease mention recognition, used for detecting the pre-identified disease mentioned, which is defined in the standard database identifiers.
- Disease concept recognition, used for identifying the diseases mentioned in the biomedical text that have not been pre-identified.

8 DISCUSSION AND CONCLUSION

In this paper, a survey was presented that discusses state-of-art language models. It can be seen that with the recent developments in neural network models, numerous types of neural language models have been proposed. Each model comes with its own advantages and disadvantages. This survey is mainly about the different types of language models, their architecture, why a model was proposed, and the datasets that were used to pre-train and fine-tune the language models.

The survey discusses how each article was selected considering the number of citations of the article, the year of publication, and the h-index of the venue. The name of the authors, from the selected articles, was used to form a word cloud to represent the authors currently working in the field of language model.

The survey focuses on the recent language models such as ELMo and BERT, and this is highlighted in Fig. 6. Section 3 discusses the classification of word embedding and from Fig. 6 it is seen that word embedding is categorized into two types, static word embeddings, and contextualized word embeddings.

There are three types of static word embeddings word2vec, GLoVe and fasttext. Word2vec relies on local information of language and captures the linguistic context of a text by clustering similar words together. GLoVe, however, tries to capture the context using both local and global information. On the other hand, *fasttext* creates a word representation using *n-grams* models, where each word is represented as a bag of character *n-grams*. One of the main forms of contextualized word embeddings is neural language model, which has been introduced to address the issue of data sparsity. Neural language models, such as ELMo, BERT, BioBERT, use a large volume of pre-trained word representations. ELMo, a feature based model, was introduced to consider the syntactic characteristic and ambiguity of a text. BERT, on the other hand, uses a neural network approach for word representations. The advantage of using BERT is that it applies bidirectional transformer language model and this helps BERT to stick to the context of a text. BERT was pre-trained on a general domain, as a result, other models similar to BERT but pre-trained on different domain corpus were released. For example, BioBERT has the same structure as BERT but it was pre-trained on biomedical corpus for biomedical text analysis. Similarly, to extract information from scientific text SciBERT was released.

This survey article also includes different datasets commonly used in the field of language models for pre-training or fine tuning a model to increase the accuracy for a particular NLP task. When BERT was introduced it was fine-tuned on a number of datasets, such as RACE, SQuAD, and GLUE, to compare the accuracy of BERT with the existing language models. Both RACE and SQuAD are question answering datasets. SQuAD contains more than 100,000 questions answered by crowdworkers and it was constructed in such a way that the model has to predict the answer from the context of the passage. Likewise, RACE dataset has several questions and each question has a set of four answers. It was designed in such a way that to answer the questions critical thinking is necessary. Thus, accessing a model's capability to understand a text.

Overall this survey provides a detailed discussion of the recent work done in the field of language model and would be a good base for researches who are new in this field or intends to produce new models for text analysis.

ACKNOWLEDGEMENT

This research was funded by Natural Sciences and Engineering Research Council (NSERC), Canada. The authors would like to thank Abhijit Rao, Dhivya Chandrasekaran and Kazi Khanam for proof reading this survey paper.

REFERENCES

[1] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," *arXiv preprint arXiv:1704.05426*, 2017.

[2] M. E. Peters, W. Ammar, C. Bhagavatula, and R. Power, "Semi-supervised sequence tagging with bidirectional language models," *arXiv preprint arXiv:1705.00108*, 2017.

[3] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[4] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," *arXiv preprint arXiv:1705.02364*, 2017.

[5] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[6] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[7] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," in *Advances in neural information processing systems*, 2015, pp. 3079–3087.

[8] F. Pittke, H. Leopold, and J. Mendling, "Automatic detection and resolution of lexical ambiguity in process models," *IEEE Transactions on Software Engineering*, vol. 41, no. 6, pp. 526–544, 2015.

[9] J. Shang, J. Liu, M. Jiang, X. Ren, C. R. Voss, and J. Han, "Automated phrase mining from massive text corpora," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 10, pp. 1825–1837, 2018.

[10] S. Y., L. Z., and H. C. W. . C. A., "Neural language modeling by jointly learning syntax and lexicon," *arXiv*, 2017.

[11] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "Semantic textual similarity-multilingual and cross-lingual focused evaluation," 2017.

[12] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[13] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[15] G. Wiese, D. Weissenborn, and M. Neves, "Neural domain adaptation for biomedical question answering," in *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, 2017, pp. 281–289.

[16] M. Habibi, L. Weber, M. Neves, D. L. Wiegandt, and U. Leser, "Deep learning with word embeddings improves biomedical named entity recognition," *Bioinformatics*, vol. 33, no. 14, pp. i37–i48, 2017.

[17] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, W. Redmond, and M. B. McDermott, "Publicly available clinical bert embeddings," *NAACL HLT 2019*, p. 72, 2019.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[19] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2249–2255.

[20] A. Mujika, F. Meier, and A. Steger, "Fast-slow recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 5915–5924.

[21] A. M. Rush, "A neural attention model for sentence summarization." *empirical methods in natural language processing*, 2015.

[22] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, 2017.

[23] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi, "Swag: A large-scale adversarial dataset for grounded commonsense inference," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 93–104.

[24] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *International conference on machine learning*, 2015, pp. 957–966.

[25] e. a. Moya, Ignacio, "An agent-based model for understanding the influence of the 11-m terrorist attacks on the 2004 spanish elections." *Knowledge-Based Systems*, 2017.

[26] J. J. Lastra-Díaz, "A reproducible survey on word embeddings and ontology-based methods for word similarity: linear combinations outperform the state of the art." *Engineering Applications of Artificial Intelligence*, 2019.

[27] B. Piotr, G. E., A. Joulin, and T. Mikolov, "Enriching word vectors with subword information." *Transactions of the Association for Computational Linguistics*, 2017.

[28] M. Giatsoglou, M. G. Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis, and K. C. Chatzisavvas, "Sentiment analysis leveraging emotions and word embeddings," *Expert Systems with Applications*, vol. 69, pp. 214–224, 2017.

[29] C.-C. Jose and M. T. Pilehvar, "From word to sense embeddings: A survey on vector representations of meaning." *Journal of Artificial Intelligence Research*, 2018.

[30] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[31] O. Levy and Y. Goldberg, "Dependency-based word embeddings," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 302–308.

[32] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun, "Topical word embeddings," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[33] O. Levy, Y. Goldberg, and I. Dagan, "Improving distributional similarity with lessons learned from word embeddings," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 211–225, 2015.

[34] T. Schnabel, I. Labutov, D. Mimno, and T. Joachims, "Evaluation methods for unsupervised word embeddings," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 298–307.

[35] E. M. Y. Finkelstein L.and Gabrilovich, R. E. S. Z., W. G., and R. E, "Placing search in context: The concept revisited." *Proceedings of the 10th international conference on World Wide Web*, 2015.

[36] H. Hua and J. Lin, "Pairwise word interaction modeling with deep neural networks for semantic similarity measurement." *Association for Computational Linguistics*, 2016.

[37] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," in *Advances in neural information processing systems*, 2016, pp. 4349–4357.

[38] X. Chen, L. Xu, Z. Liu, M. Sun, and H. Luan, "Joint learning of character and word embeddings," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[39] X. Wang, A. McCallum, and X. Wei, "Topical n-grams: Phrase and topic discovery, with an application to information retrieval," in *Seventh IEEE international conference on data mining (ICDM 2007)*. IEEE, 2007, pp. 697–702.

[40] T. Kenter and M. De Rijke, "Short text similarity with word embeddings," in *Proceedings of the 24th ACM international on conference on information and knowledge management*, 2015, pp. 1411–1420.

[41] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández, "Syntactic n-grams as machine learning features for natural language processing," *Expert Systems with Applications*, vol. 41, no. 3, pp. 853–860, 2014.

[42] M. T, K. S., B. L., J. Černocký, and K. S., "Extensions of recurrent neural network language model," *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2011.

[43] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[44] G. D, R. D., M. Mitra, and . J. G. J., "Word embedding based generalized language model for information retrieval," *ACM SIGIR conference on research and development in information retrieval*, 2015.

[45] N. A. T. and T. N. Nguyen, "Graph-based statistical language model for code," *IEEE International Conference on Software Engineering*, 2015.

[46] 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*, 2017, pp. 5998–6008.

[47] L. Yuhua, Z. A. Bandar, and D. McLean, "An approach for measuring semantic similarity between words using multiple information sources." *IEEE Transactions on knowledge and data engineering 15.4*, 2003.

[48] Z. Yang, "Breaking the softmax bottleneck: A high-rank rnn language model." *arXiv*, 2017.

[49] R. Das, M. Zaheer, and C. Dyer, "Gaussian lda for topic models with word embeddings," 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)*, 2015, pp. 795–804.

[50] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[51] L. Liu, J. Shang, X. Ren, F. F. Xu, H. Gui, J. Peng, and J. Han, "Empower sequence labeling with task-aware neural language model," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[52] I. Iacobacci, M. T. Pilehvar, and R. Navigli, "Embeddings for word sense disambiguation: An evaluation study," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 897–907.

[53] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 1998, pp. 604–613.

[54] C. Li, H. Wang, Z. Zhang, A. Sun, and Z. Ma, "Topic modeling for short texts with auxiliary word embeddings," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 165–174.

[55] D. Shen, Y. Zhang, R. Henao, Q. Su, and L. Carin, "Deconvolutional latent-variable model for text sequence matching," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[56] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[57] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[58] W. A and C. K., "Bert has a mouth, and it must speak: Bert as a markov random field language model." *IEEE Spoken Language Technology Workshop*, 2019.

[59] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," in *Advances in Neural Information Processing Systems*, 2017, pp. 6294–6305.

[60] A. Mnih and G. E. Hinton, "A scalable hierarchical distributed language model," in *Advances in neural information processing systems*, 2009, pp. 1081–1088.

[61] R. I. Doğan, A. Névéol, and Z. Lu, "A context-blocks model for identifying clinical relationships in patient records," *BMC bioinformatics*, vol. 12, no. S3, p. S3, 2011.

[62] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[63] R. I. Doğan, R. Leaman, and Z. Lu, "Ncbi disease corpus: a resource for disease name recognition and concept normalization," *Journal of biomedical informatics*, vol. 47, pp. 1–10, 2014.

[64] W. Yoon, C. H. So, J. Lee, and J. Kang, "Collabonet: collaboration of deep neural networks for biomedical named entity recognition," *BMC bioinformatics*, vol. 20, no. 10, p. 249, 2019.

[65] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.

[66] I. Beltagy, A. Cohan, and K. Lo, "Scibert: Pretrained contextualized embeddings for scientific text," *arXiv preprint arXiv:1903.10676*, 2019.

[67] Y. Belinkov and J. Glass, "Analysis methods in neural language processing: A survey," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 49–72, 2019.

[68] J. Kringelum, S. K. Kjaerulff, S. Brunak, O. Lund, T. I. Oprea, and O. Taboureau, "Chemprot-3.0: a global chemical biology diseases mapping," *Database*, vol. 2016, 2016.

[69] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii, "Genia corpus—a semantically annotated corpus for bio-textmining," *Bioinformatics*, vol. 19, no. suppl_1, pp. i180–i182, 2003.

[70] M. Neumann, D. King, I. Beltagy, and W. Ammar, "Scispacy: Fast and robust models for biomedical natural language processing," *arXiv preprint arXiv:1902.07669*, 2019.

[71] B. Nye, J. J. Li, R. Patel, Y. Yang, I. J. Marshall, A. Nenkova, and B. C. Wallace, "A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature," in *Proceedings of the conference. Association for Computational Linguistics. Meeting*, vol. 2018. NIH Public Access, 2018, p. 197.

[72] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual bert?" *arXiv preprint arXiv:1906.01502*, 2019.

[73] J. Li, Y. Sun, R. J. Johnson, D. Sciaky, C.-H. Wei, R. Leaman, A. P. Davis, C. J. Mattingly, T. C. Wiegers, and Z. Lu, "Biocreative v cdr task corpus: a resource for chemical disease relation extraction," *Database*, vol. 2016, 2016.

[74] W. Ammar, D. Groeneveld, C. Bhagavatula, I. Beltagy, M. Crawford, D. Downey, J. Dunkelberger, A. Elgohary, S. Feldman, V. Ha *et al.*, "Construction of the literature graph in semantic scholar," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, 2018, pp. 84–91.

[75] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[76] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. Liu, M. Peters, M. Schmitz, and L. Zettlemoyer, "Allennlp: A deep semantic natural language processing platform," *arXiv preprint arXiv:1803.07640*, 2018.

[77] K. Huang, J. Altosaar, and R. Ranganath, "Clinicalbert: Modeling clinical notes and predicting hospital readmission," *arXiv preprint arXiv:1904.05342*, 2019.

[78] A. Cohan, W. Ammar, M. van Zuylen, and F. Cady, "Structural scaffolds for citation intent classification in scientific publications," *arXiv preprint arXiv:1904.01608*, 2019.

[79] T. Dozat and C. D. Manning, "Deep biaffine attention for neural dependency parsing," *arXiv preprint arXiv:1611.01734*, 2016.

[80] Z. C. and L. J, "A study of smoothing methods for language models applied to ad hoc information retrieval," *ACM SIGIR*, 2017.

[81] O. Melamud, J. Goldberger, and I. Dagan, "context2vec: Learning generic context embedding with bidirectional lstm," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 51–61.

[82] S. M, N. H, and S. R., "From feedforward to recurrent lstm neural networks for language modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2015.

[83] W. Liu, R. Lin, Z. Liu, L. Liu, Z. Yu, B. Dai, and L. Song, "Learning towards minimum hyperspherical energy," in *Advances in neural information processing systems*, 2018, pp. 6222–6233.

[84] D. Yang, W. Wei, L. Wang, and Hon, "Unified language model pre-training for natural language understanding and generation," *In Advances in Neural Information Processing Systems*, 2019.

[85] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[86] E. S, B. A., and A. M., "Pre-trained language model representations for language generation," *arXiv*, 2015.

[87] T. S., K. A., C. C. C., W. Y., S. T. N., and L. K., "A comparison of techniques for language model integration in encoder-decoder speech recognition," *In 2018 IEEE Spoken Language Technology Workshop*, 2018.

[88] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," pp. 632–642, 2015.

[89] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.

[90] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[91] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato, "Unsupervised machine translation using monolingual corpora only," *arXiv preprint arXiv:1711.00043*, 2017.

[92] D. Jia, D. Wei, S. Richard, L. Li-Jia, L. Kai, and F.-F. Li, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition.* Ieee, 2009, pp. 248–255.

[93] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[94] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "Race: Large-scale reading comprehension dataset from examinations," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 785–794.

[95] W. Wang, M. Yan, and C. Wu, “Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1705–1714.

[96] M. Richardson, C. J. Burges, and E. Renshaw, “Mctest: A challenge dataset for the open-domain machine comprehension of text,” in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 193–203.

[97] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, “Exploring the limits of language modeling,” *Advances in Neural Information Processing Systems*, 2016.

[98] A. Warstadt, A. Singh, and S. R. Bowman, “Neural network acceptability judgments,” *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 625–641, 2019.

[99] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[100] Y. Yang, S. Yuan, D. Cer, S.-y. Kong, N. Constant, P. Pilar, H. Ge, Y.-H. Sung, B. Strope, and R. Kurzweil, “Learning semantic textual similarity from conversations,” in *Proceedings of The Third Workshop on Representation Learning for NLP*, 2018, pp. 164–174.

[101] J. G. Mork, O. Bodenreider, D. Demner-Fushman, R. I. Doğan, F.-M. Lang, Z. Lu, A. Névéol, L. Peters, S. E. Shooshan, and A. R. Aronson, “Extracting rx information from clinical narrative,” *Journal of the American Medical Informatics Association*, vol. 17, no. 5, pp. 536–539, 2010.

[102] J. P. Chiu and E. Nichols, “Named entity recognition with bidirectional lstm-cnns,” *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 357–370, 2016.

[103] K. S. Tai, R. Socher, and C. D. Manning, “Improved semantic representations from tree-structured long short-term memory networks,” *arXiv preprint arXiv:1503.00075*, 2015.

[104] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, “Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1601–1611.

[105] J. M. Giorgi and G. D. Bader, “Transfer learning for biomedical named entity recognition with neural networks,” *Bioinformatics*, vol. 34, no. 23, pp. 4087–4094, 2018.

[106] Y. Zeng, Y. Feng, R. Ma, Z. Wang, R. Yan, C. Shi, and D. Zhao, “Scale up event extraction learning via automatic training data generation,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

A TABLE OF REFERENCES

Table 5. References selected for the survey, including total number of citations, h-Google Index of the venue and year of publication

Title	Venue	Total no. of Citation as of April'20	h-Google index	Year of publication
BioBERT: a pre-trained biomedical language representation model for biomedical text mining [62]	Bio Informatics	141	335	2018
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [18]	arXiv	4500		2018
A Neural Probabilistic Language Model [12]	Journal of Machine Learning Research	5862	173	2003
Recurrent neural network based language model [13]	Eleventh annual conference of the international speech communication association	3949	65	2010
Improved semantic representations from tree-structured long short-term memory networks [103]	Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics	1604	106	2015
Deep contextualized word representations[56]	arXiv	2424		2018
A large annotated corpus for learning natural language inference [88]	Empirical Methods in Natural Language Processing	999	88	2015
Semi-supervised sequence learning[7]	Advances in neural information processing systems	542	169	2015
Universal Language Model Fine-tuning for Text Classification [5]	Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics	620	106	2015
Imagenet: A large-scale hierarchical image database [92]	2009 IEEE conference on computer vision and pattern recognition	16242	240	2009
Empower sequence labeling with task-aware neural language model [51]	AAAI Conference on Artificial Intelligence	127	95	2018
Distributed representations of words and phrases and their compositionality [14]	Neural Information Processing Systems	18000	169	2013

Continued on next page

Table 5 – *Continued from previous page*

Title	Venue	Total no. of Citation as of April'20	h-Google index	Year of publication
GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding [89]	Empirical Methods in Natural Language Processing	312	88	2018
Exploring the Limits of Language Modeling [97]	arXiv	649		2018
SQuAD: 100,000+ Questions for Machine Comprehension of Text [6]	Empirical Methods in Natural Language Processing	1365	88	2016
Language models are unsupervised multitask learners [50]	Open AI Blog	317		2019
Semi-supervised sequence tagging with bidirectional language models [2]	Association for Computational Linguistics	239	106	2017
Deep convolutional neural network for inverse problems in imaging [22]	ieee transactions on image processing	587	242	2017
Unsupervised machine translation using monolingual corpora only [91]	arXiv	261		2017
Glove: Global vectors for word representation [43]	Empirical Methods in Natural Language Processing	12000	88	2014
Attention is all you need [46]	Neural Information Processing Systems	6000	169	2017
TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension [104]	Association for Computational Linguistics	262	106	2017
How transferable are features in deep neural networks? [93]	Neural Information Processing Systems	3677	169	2014
A Decomposable Attention Model for Natural Language Inference [19]	Association for Computational Linguistics	529	106	2016
A scalable hierarchical distributed language model [60]	Neural Information Processing Systems	5862	173	2003
Context2vec: Learning generic context embedding with bidirectional lstm [81]	Computational Natural Language Learning	191	39	2016

Continued on next page

Table 5 – *Continued from previous page*

Title	Venue	Total no. of Citation as of April'20	h-Google index	Year of publication
A theoretically grounded application of dropout in recurrent neural networks [85]	Neural Information Processing Systems	843	169	2016
Learning towards minimum hyperspherical energy [83]	Neural Information Processing Systems	587	169	2016
Fast-slow recurrent neural networks [20]	Neural Information Processing Systems	39	169	2019
Transfer learning for biomedical named entity recognition with neural networks [105]	Bio Informatics	27	335	2018
Collabonet: collaboration of deep neural networks for biomedical named entity recognition [64]	BMC Bio Informatics	10	335	2019
Neural domain adaptation for biomedical question answering [15]	Association for Computational Linguistics	32	183	2017
Publicly available clinical bert embeddings [17]	Association for Computational Linguistics	45	183	2019
NCBI disease corpus: a resource for disease name recognition and concept normalization [63]	Journal of biomedical informatics	201	83	2014
Deep learning with word embeddings improves biomedical named entity recognition [16]	Bio Informatics	192	335	2017
Supervised Learning of Universal Sentence Representations from Natural Language Inference Data [4]	arXiv	680		2018
Learned in translation: Contextualized word vector [59]	Neural Information Processing Systems	360	169	2017
State-of-the-Art Speech Recognition with Sequence-to-Sequence Models [3]	IEEE International Conference on Acoustics, Speech and Signal Processing	376	130	2018
Automatic Detection and Resolution of Lexical Ambiguity in Process Models [8]	IEEE Transactions on Software Engineering	46	151	2015
Named entity recognition with bidirectional LSTM-CNNs [102]	transactions of the Association for Computational Linguistics	743	51	2016

Continued on next page

Table 5 – *Continued from previous page*

Title	Venue	Total no. of Citation as of April'20	h-Google index	Year of publication
Automated phrase mining from massive text corpora [9]	IEEE Transactions on Knowledge and Data Engineering	86	77	2018
LSwag: A large-scale adversarial dataset for grounded commonsense inference [23]	Empirical Methods in Natural Language Processing	90	157	2018
A broad-coverage challenge corpus for sentence understanding through inference [1]	Association for Computational Linguistics	378	183	2018
Multigranularity hierarchical attention fusion networks for reading comprehension and question answering [95]	Association for Computational Linguistics	62	183	2018
Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation [11]	Association for Computational Linguistics	210	183	2018
A context-blocks model for identifying clinical relationships in patient records [61]	BMC bioinformatics	51	183	2011
Extracting Rx information from clinical narrative [101]	JAMIA	31	132	2010
ALBERT: A Lite BERT for Self-supervised Learning of Language Representations [65]	arXiv	48		2019
RACE: Large-scale ReAding Comprehension Dataset From Examinations [94]	Empirical Methods in Natural Language Processing	169	335	2017
Improving language understanding by generative pre-training [90]	OPEN AI	680		2018
Neural Network Acceptability Judgments [98]	Transactions of the Association for Computational Linguistics	57	80	2019
Recursive deep models for semantic compositionality over a sentiment treebank [99]	Empirical Methods in Natural Language Processing	3596	335	2013
Learning Semantic Textual Similarity from Conversations [100]	arXiv	46		2018
Scibert: Pretrained contextualized embeddings for scientific text [66]	arXiv	53		2019

Continued on next page

Table 5 – *Continued from previous page*

Title	Venue	Total no. of Citation as of April'20	h-Google index	Year of publication
Analysis methods in neural language processing: A survey [67]	Transactions of the Association for Computational Linguistics	43	168	2019
Construction of the Literature Graph in Semantic Scholar [74]	Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)	44	51	2018
Google's neural machine translation system: Bridging the gap between human and machine translation [75]	arXiv	2370		2016
Deep biaffine attention for neural dependency parsing [79]	arXiv	243		2016
GENIA corpus—a semantically annotated corpus for bio-textmining [69]	Bioinformatics	1024	183	2003
Scispacy: Fast and robust models for biomedical natural language processing [70]	arXiv	20		2019
A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature [71]	Proceedings of the conference. Association for Computational Linguistics. Meeting	20		2018
ChemProt-3.0: a global chemical biology diseases mapping	Database: The Journal of Biological Databases and Curation [68]	40	65	2016
Allennlp: A deep semantic natural language processing platform [76]	arXiv	229		2018
Structural scaffolds for citation intent classification in scientific publications [78]	arXiv	7		2019
Clinicalbert: Modeling clinical notes and predicting hospital readmission [77]	arXiv	17		2019
BioCreative V CDR task corpus: a resource for chemical disease relation extractions [73]	Database: The Journal of Biological Databases and Curation	88	65	2016
Efficient Estimation of Word Representations in Vector Space [30]	arXiv	14538		2013

Continued on next page

Table 5 – *Continued from previous page*

Title	Venue	Total no. of Citation as of April’20	h-Google index	Year of publication
Dependency-based word embeddings [31]	Transactions of the Association for Computational Linguistics	850	168	2014
Topical word embeddings [32]	AAAI	280	153	2015
From word embeddings to document distances [24]	International conference on machine learning	940	254	2015
Improving distributional similarity with lessons learned from word embeddings [33]	Association for Computational Linguistics	923	168	2014
Evaluation methods for unsupervised word embeddings [34]	Association for Computational Linguistics	334	335	2017
Google’s neural machine translation system: Bridging the gap between human and machine translation [75]	arXiv	2370		2016
BioCreative V CDR task corpus: a resource for chemical disease relation extractions c[73]	Database: The Journal of Biological Databases and Curation	88	65	2016
Approximate nearest neighbors: towards removing the curse of dimensionality [53]	Proceedings of the thirtieth annual ACM symposium on Theory of computing	4356	89	1998
How multilingual is Multilingual BERT? [72]	arXiv	40		2019
Scale up event extraction learning via automatic training data generation [106]	AAAI	8	95	2018
Enriching word vectors with subword information [27]	Transactions of the Association for Computational Linguistics	3000	183	2017
From word to sense embeddings: A survey on vector representations of meaning [29]	Journal of Artificial Intelligence Research	71	103	2018
An agent-based model for understanding the influence of the 11-M terrorist attacks on the 2004 Spanish elections [25]	Knowledge-Based Systems	7	94	2019
Pairwise word interaction modeling with deep neural networks for semantic similarity measurement [36]	Association for Computational Linguistics	140	183	2016

Continued on next page

Table 5 – *Continued from previous page*

Title	Venue	Total no. of Citation as of April'20	h-Google index	Year of publication
An approach for measuring semantic similarity between words using multiple information sources [47]	IEEE Transactions on knowledge and data engineering	1315	148	2003
Breaking the softmax bottleneck: A high-rank RNN language model [48]	arXiv	178		2017
A neural attention model for sentence summarization [21]	Empirical Methods in Natural Language Processing	1359	103	2015
Placing search in context: The concept revisited [35]	international conference on World Wide Web	75	64	2016
A reproducible survey on word embeddings and ontology-based methods for word similarity: linear combinations outperform the state of the art [26]	Association for Computational Linguistics	7	183	2019
Extensions of recurrent neural network language model [42]	ACM SIGIR	78	65	2016
Graph-based statistical language model for code [45]	IEEE International Conference on Software Engineering	42	103	2016
A study of smoothing methods for language models applied to ad hoc information retrieval [80]	ACM SIGIR	53	81	2016
Unified language model pre-training for natural language understanding and generation	In Advances in Neural Information Processing Systems	42	75	2015
A comparison of techniques for language model integration in encoder-decoder speech recognition [87]	IEEE Spoken Language Technology Workshop	229		2018
From feedforward to recurrent LSTM neural networks for language modeling [82]	arXiv	7		2019
Pre-trained language model representations for language generation [86]	Association for Computational Linguistics	20	163	2019
Neural language modeling by jointly learning syntax and lexicon [10]	arXiv	38		2017