Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7688125

AN ANALYSIS OF WORD EMBEDDING MODELS WIDE-RANGING OF SOTA TRANSFORMERS

T. Priyanka
Research Scholar, Department of CS&SE, Andhra University College of Engineering (A), Visakhapatnam, Andhra Pradesh, Pin – 530003, INDIA.
sripadarsau@gmail.com

A. Mary Sowjanya
Associate Professor, Department of CS&SE, Andhra University College of Engineering (A), Visakhapatnam, Andhra Pradesh, Pin – 530003, INDIA.
sowmaa@gmail.com,

Abstract
Research on word representation has always been an important area of interest in the antiquity of Natural Language Processing (NLP). Interpreting such intricate linguistic data is essential, since it carries a wealth of information and is useful for many applications. In the context of NLP, Deep Learning manifests as word embeddings, which are extensively used to represent words of a document as multi-dimensional numeric vectors in place of traditional word representations. In deep learning models, word embeddings are crucial part of providing input features for downstream tasks, such as sequence labeling, text classification etc., large amounts of text can be converted into effective vector representations that capture the same semantic information using these approaches. Furthermore, several learning algorithms can use such representations for a range of NLP-related tasks. The effectiveness or accuracy of an embedding can be established if it could be transferred to a downstream task of NLP to surpass the performance levels that could be reached by traditional machine learning algorithms. Over the past decade a number of word embedding methods, mainly catered to the traditional and context-based categories, were proposed. As part of this study, we examine different word representation models in terms of their power of expression, from historical models to today's state-of-the-art word representation language models.
Keywords: *NLP, Machine Learning, word embedding, Deep Learning, Language model.*

1. Introduction

Recent advancements in machine learning have resulted in unprecedented success in various applications related to Natural Language Processing (NLP). Specifically, the spectrum of NLP applications ranging from topic-wise text classification, sentiment analysis of customer reviews, document clustering in search engines or other web-based information retrieval systems, question answering systems, co-reference resolution, semantic role labeling, named entity recognition, textual entailment/inference, document summarization and machine translation of text from one language to the other. While simpler of these applications could be handled by the traditional machine learning algorithms, the complex applications demand

for the recent advancements of ML towards Deep Learning (DL) and Transfer Learning (TL) as they have proven ability for complex feature extraction and object representation.

Deep Learning involves stacked neural networks for extracting hierarchy of features from complex objects. It provides a solution to alleviate the problem of exploding or vanishing gradients as more number of hidden layers are required for extracting complex features of objects in a feed-forward Neural Network. Different types of DL network architectures were developed namely, Restricted Boltzmann machines, Auto-encoders, Convolution Neural Networks, Recurrent Neural Networks amidst Gated Recurrent Units and Long Short Term Memory cells to deal with a wide variety of applications. For example, CNNs are suitable for extracting structural/spatial features based on the locality from images, while LSTM cells are suitable for extracting features from sequential objects. However, DL approaches require enormously large amounts of labeled data for building successful models. Availability of huge amounts of labeled data may not always be possible and hence becomes a bottle-neck for the applicability of DL for supervised model building in domains with scarce labeled examples. Such applications can be successfully handled with Transfer Learning where in, the knowledge acquired in the process of model building from a labeled/unlabeled data in source domain is used to build an effective supervised model with less labeled data in target domain. The features extracted from enormously large data sets are fine-tuned or tailor-made for the specific task as defined in the target domain with limited labeled data.

Textual data is quickly rising, with unstructured text with lower quality generating at a faster rate than structural text. Textual data can be found in a range of areas, including social media, online forums, articles published, patient's clinical reports, and user reviews where people voice their opinions about certain products or services. Post text pre-processing, feature extraction is a vital step. Searching for an article on the trending topics results in hundreds of results, identifying the sentiment of customer reviews for a service, company or a product, recommending the movies depending on user watch history horror, comedy etc., translating the sentences to our required language. These all applications are dealing with enormous amount of text. As textual data cannot be provided as input directly to the machine learning and deep learning models, they have to be represented as numerical values. Here word embeddings comes into picture.

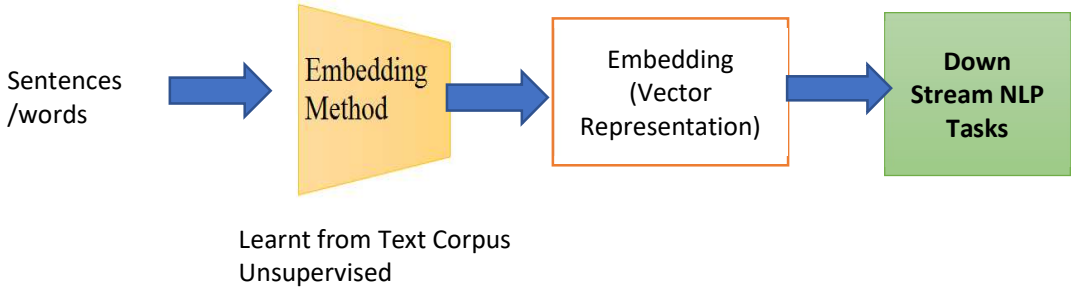

Fig 1: Word embedding

In NLP, TL takes the form of word vectors, which are widely used to encode terms in a document using multi-dimensional integer vectors rather than standard word interpretations

like one hot vectors, TF-IDF vectors, etc., The examples are GloVe and Word2Vec. They capture the semantics of a word and place the words with similar meaning in close proximity. However, they are context independent and hence cannot handle polysemy. The later development in the form of CoVe dynamically learns contextualized word embeddings from large collection of text using biLSTM in support of machine translation. However, it represents the words based only on the final layer of the model, leaving out the information from the lower layers. It was found that in a multi-layered LSTM architecture the outer layers capture task-specific features, while the inner layers capture more generic/basic features which would be the building blocks for more specific features. The recent advancement on Word embeddings proposed in ELMo are multi-layered and hence has better potential for getting fine-tuned with appropriate weights for different layers to suit to a wide variety of NLP applications. For example, the textual entailment relies more on task specific embeddings and hence gives higher weightage to outer layers, while POS tagging relies more on the basic features and gives higher weightage to inner layers. The word embeddings provided by ELMo are considered as Universal Embeddings, since the generic Language Model (LM) once pre-trained, can be used for a wide variety of NLP tasks with appropriate fine-tuning.

Word embeddings have limitations in natural language comprehension as it is unnatural to understand text, word-by-word rather than larger chunks of texts, such as, phrases, sentences and paragraphs [43]. Sentence embeddings, if learned accurately, are more semantically robust than word embeddings as they capture a holistic essence of the sentence. In general, sentence embeddings are formed based on the embeddings of the constituent words. For example, the ELMo word embeddings may be used to form Sentence embeddings. SkipThought, QuickThoughts, Sent2Vec, Universal Sentence Encoder are some of the methods that derive the sentence embeddings directly. However, representation of text using sentence embeddings is not as successful as that of word embeddings for the present. Recently, [22] explored Seven different architectures for directly extracting sentence embeddings from SNLI data set and proof shown that biLSTM along with maxpooling can secure good via supervised learning.
Word representation model is based on word recurrence. A text is translated into a vector form that contains the number of words that appear in a document using those word count approaches. Researchers can use word embeddings to capture significant information derived from the context of the words, or any other word. One of the most common ways to express vocabulary in a document is by word embedding. This can recognise a word's surroundings in a document, semantics and grammatical similarity, and relationships with other words, among other. Words with equivalent meanings that occur repeatedly in similar contexts have near spatial locations. The fundamental idea is to keep context data into word vectors. Any cosine angle made by these vectors ought to be near to 1, i.e. angle should be close to 0.

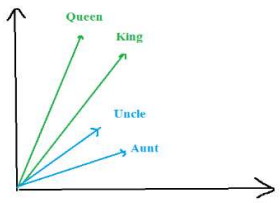

Fig 2: Words having similar meaning or relationship are closer in the vector space

This allows a computer system to better comprehend word meanings, which is effective for a range of applications. Word embeddings, for example, can be utilised to produce more accurate translations between languages or to summarise extensive texts. Researcher employed embeddings to determine sentiment (emotional states) and to categorise articles into certain groups (e.g., news or sports or entertainment).

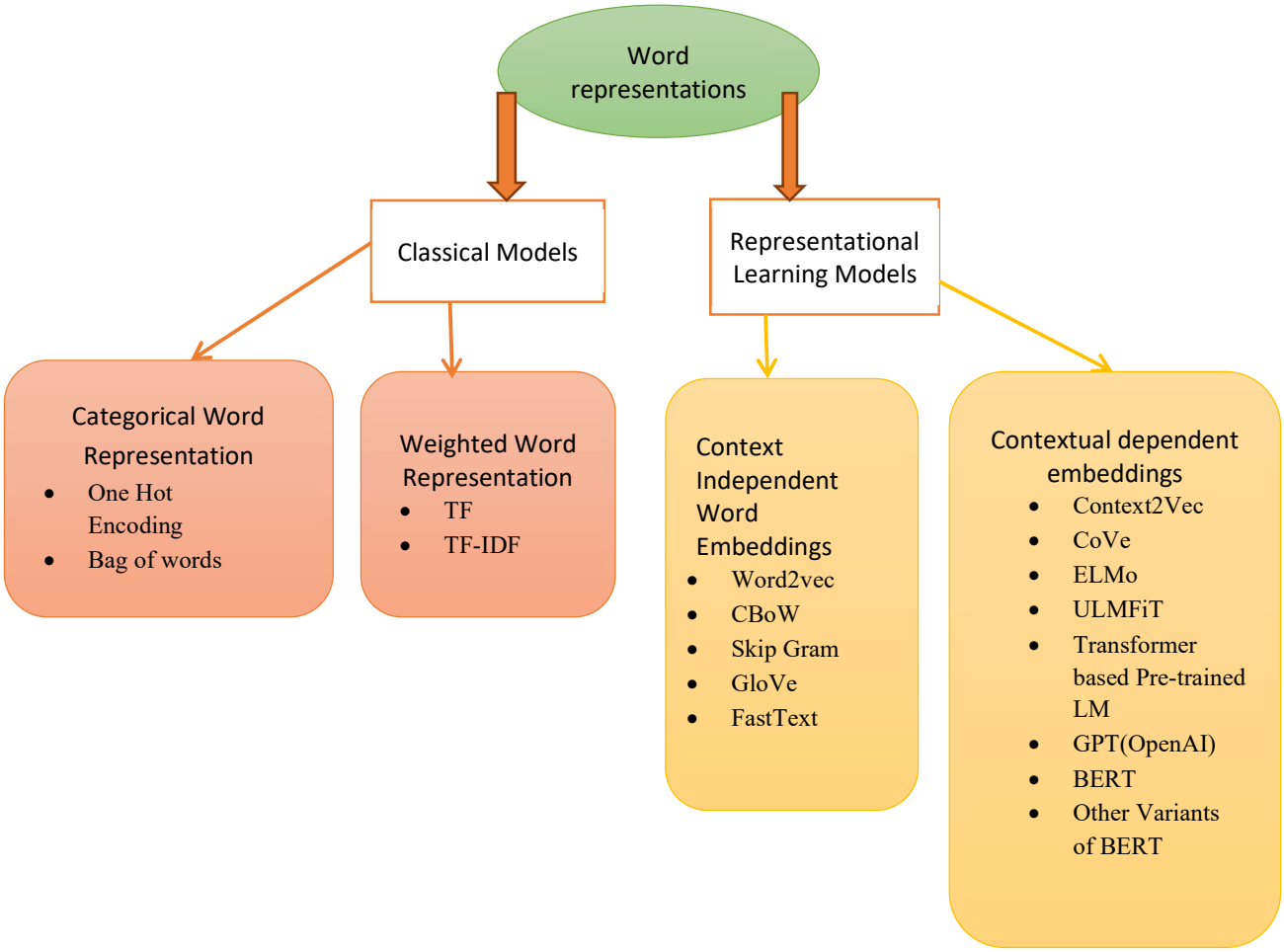

Fig 3: Categorization of Word Embeddings

3. Related Work

3.1 Traditional Models

This section discusses some of the traditional text classification models that were popular recently. This form of word representation method is based on word frequency. A text is translated into a vector form that reflects the number of words that occurs in a document using these approaches. We begin with a brief overview of categorical word representation approaches, followed by a discussion of weighted word representation methods.

3.1.1 Categorical word representation is the most basic technique of representing text. Words are represented by a symbol either "1" or "0" in this manner. The two models that fall under categorical word representation approaches are one-hot encoding and Bag-of-words (BoW). Both are discussed briefly here.

One hot encoding: One hot encoding[31] is the most straightforward technique of text representation. The dimension of one hot encoding is the same as the number of terms in the vocabulary. Every word in the vocabulary is represented by a binary value as 0 or 1. This means that every word is comprised entirely of ones and zeros.Create a zero vector with a dimension identical to vocabulary, and put one in the position that refers to the word for representing each word. The relevant word's index is indicated with 1, while all others are given a number 0. Every unique word will have one dimension, in that dimension; it will be represented by a 1and with 0s everywhere else. This method is not efficient due to its sparsity.

- **Bag-of-Words (BoW):**[1]An advancement technique of one-hot encoding. The BOW [32] approach is utilized in a variety of fields, including computational linguistics (NLP), computer vision (CV), and information extraction (IR), and others. The semantic relationships between words, as well as the order of words and grammar are omitted in the matrix of words created with BOW. As previously stated, BOW is a one-hot encoding extension that encodes words as a 1-hot-encoded column vector in the lexicon. The size of the vectors increases as the vocab increases. Moreover, a great amount of "0s" can lead to a shallow matrix containing no data order.Illustration of one-hot encoding and Retrieving Features is shown in fig.4.

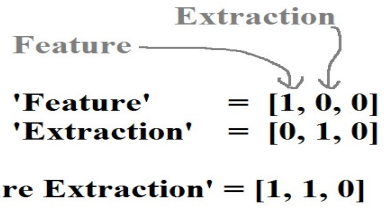

Fig 4: An interpretation of one-hot encoding and BoW methods

3.1.2 Representing Words using Weights

TF and TF-IDF are two common approaches for weighted word representations. Below is a brief discussion of both of them.

- **Term Frequency:** The general approach of text feature extraction is term frequency, measured as count of each word occurrence in a document divided by document length.
- **Term Frequency-Inverse Document Frequency (TF-IDF):** [2] A technique introduced that can be used with TF to reduce the influence of common phrases. by assigning less weights, so that rarely occurred words having more weight implies they are significant in the document. [33, 34] The orders of words in a document, as well as the semantics and syntax information of words, are not captured. This is appropriate for usage as a lexical level feature.

$$TF - IDF(w, d, D) = TF(w, d) * log(\frac{D}{ds})$$

where w indicates the words/terms; d denotes a specific document; D represents the Corpus; and ds denotes count of this word occurred in D.

3.2 Representation Learning

Previous models suffer from the curse of high dimensionality and failed to capture the syntax and semantics of the words. The findings from a study revealed that learning distributed

word representation[3] in low dimension space to get rid of flaws in those models. The findings from a study revealed that learning distributed word representation in low dimension space to get rid of flaws in those models. Various approaches have been developed in the past to automatically discover representations for downstream tasks such as categorization using word embeddings [4]. Feature learning or representation learning are algorithms that discover features on their own. Before neural LM[5], hand-crafted features were mostly utilized to simulate tasks in natural language.

3.2.1 Continuous Words Representation (Non-Contextual Embeddings)

The Non-Contextual Word Embedding Model generates only one vector, ignoring the position of word in a sentence. For example, in the sentence "She strolled along the seashore after paying EMI in bank," the term "bank" has different implications based on the context where sentence occurs, but these models simply results into a single vector for 'bank'.

1. Word2vec

Word2Vec [6] is a popular shallow neural network-based approach for embedding words. Either Continuous Bag of Words (CBOW) or SkipGram are often used. Word2vec is a collection of interrelated techniques for generating text embeddings[4]. These models are shallow, two-layer neural systems that are prepared to recreate etymological settings of words. Word2vec [35] takes as its information an enormous corpus of content and creates a vector space, ordinarily of a few hundred measurements, with every novel word in the corpus being allotted a relating vector in the space. Word vectors are situated in the vector space to such an extent that words that offer basic settings in the corpus are found near each other in the space.

(a) Continuous Bag of words (CBOW)

The CBOW [7] model uses the context of every word in a text sequence to predict one word that corresponds to the surrounding words. The count of nearby words is used as window size for defining its context. These are provided as one-hot encoded vectors, and the neural network is trained while searching for the target word.

(b) Skip-Gram

[4, 7] This method is analogous to CBOW, with an exception that it learns by guessing context words for a target text, given d_i this model produces d_i-2; d_i-1; d_i+1; d_i+2.

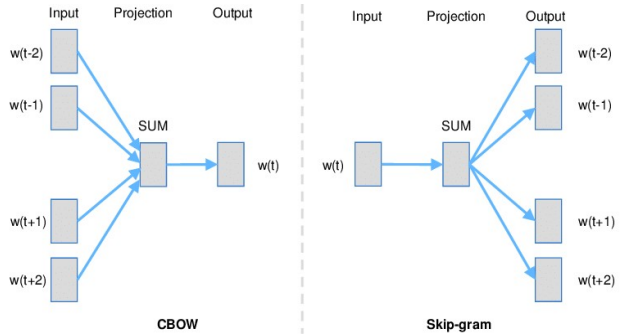

Fig 5: Working of CBOW and SkipGram [6,7]

2. Global Vectors (GloVe)

The primary idea behind the GloVe [8] word embedding is to use statistics to derive the relation between the words. GloVe has the benefit that, contrary to Word2vec, it would not depend exclusively on locally stats and word contextual knowledge to generate vector representation of each word, but also utilizes global statistical information and co-occurrence of words.It is based on word-context matrices and matrix factorization algorithms. In a huge corpus, you create a massive matrix of co-occurrence information and count each "word" (rows) and how often this word occurs in some context (columns). Typically, we inspect our corpus looking for context terms within a defined area specified by a window-size before the term and a window-size after the term, for every term. We also attach less weight to terms being farther away. Each value in the co-occurrence matrix reflects a pair of words appearing together [36]. Word embeddings are created by combining global word co-occurrence matrices from a corpus.

3. FastText

Word2Vec has a key flaw in that it does not cope with phrases which were not found in the lexicon or dictionary, whereas FastText [9] does. The analogous technique as CBOW and SkipGram employed by FastText, rather than dealing with a single word, it divides into n-grams, which then processes on those character n-grams [37]. Those summations over n-grams of characters is used to represent a word.

4. Embeddings via Transfer Learning Technique

Human seem to have natural approaches to transfer knowledge from one task to another relevant task [38]. That is, we perceive and apply important information from past learning when we experience new errands. Transfer learning [10] includes the methodology where information learned in at least one source tasks is used to improve the learning of a related objective [39]. While most AI calculations are intended to address single undertakings, the advancement of calculations that encourage Transfer learning as a point of continuous enthusiasm for the AI people group.

Table 1: Summary of Embedding Techniques

Language Model	Architecture	Pre-training Activities
CoVe	seq2seq NMT model	supervised learning via translation dataset.
ELMo	two-layer biLSTM	Predicting the next token
CVT	two-layer biLSTM	Make use of labelled and unlabelled datasets via semi-supervised learning
ULMFiT	AWD-LSTM	autoregressive pre-training on Wikitext-103

Open AI GPT	Transformer decoder	a general LM, performs classification and can predict the next token,
BERT	Transformer encoder	mask language model + Predicting the next sentence
ALBERT	Light-weight BERT	mask language model + predict order of sentence
GPT-2	Transformer decoder	Predicting the next token
RoBERTa	BERT	mask language model via dynamic masking method
T5	Transformer encoder + decoder	pre-training done with multi-task model via supervised and unsupervised tasks and for which each job is transformed into a text-to-text format
GPT-3	Transformer decoder	Predicting the next token
XLNet	same as BERT	permutation language modeling
BART	BERT encoder + GPT decoder	noised sentences were reconstructed
ELECTRA	same as BERT	Replacing the detected token

3.2.2 Word representations by using its context information

Contextual Word Embedding Models can find new word embeddings for a word which thus reflect the word's placement in a statement, making them context-dependent.

1. Generic Context word representation (Context2Vec)

Context2Vec [11] used the same concept as windowing based on the original CBOW Word2Vec model, but instead of using a simple average function, it uses a significantly more sophisticated parametric model based on one layer of Bi-LSTM in three steps [40]. They employed a Bidirectional LSTM, feeding one from left to right representations and the other from right to left context embedding representations. To convey sentential context to an embedding, a feed forward network concatenates each context vector embedding and provides a hidden representation by learning the network factors.

2. Contextualized word representations Vectors (CoVe)

To contextualise word vectors, a deep LSTM encoder with an attention sequence-to-sequence learning algorithm is employed. By training model on a Machine Translation task using a two-layer Bi-directional LSTM network and retrieved CoVe[12] from an encoder of a given task. Neural Machine Translation is viewed as a promising source for learning extremely complex representations. To translate, the model must learn to identify the deep relationships between

words in order to create an advanced representation of the entire sentence. As a result, we may assume that the NMT task encoder will preserve significantly more complicated semantic relationships between words. The concept of Transfer Learning used to train a Sequence to Sequence (seq2seq) mode for the Machine Translation. Get the seq2seq model's encoder and use it as a pre-trained layer in any task. This encoder used as a pre-trained layer in any classification or generating task after training. Each vector in GloVe(s) is concatenated with its corresponding vector in CoVe(s) [41] for classification and question answering for an input sequences.

3. Embedding from Language Models (ELMo)

The semantics of a word are defined by the context in which it is embedded, just as they are in Context2Vec. While discussing about the context of the words, there's no better way to come up with a better knowledge of the sentence representation is to use the power of Language Modelling. The goal of Language Model [5] is to create a probability distribution that represents a corpus or set of sentences. Typically, the goal of a Language Model is to guess the next word based on the preceding words. As a result, we can infer that the word is conditional on the words mentioned previously. The similar idea can be extended to the words that come after the target word. The Forward Language Model (anticipate the target word based on previous words) and a Backward Language Model (predict the target word based on subsequent words) or a Bi-directional Language Model (both Forward and Backward processes in Language Model). ELMo[13] is based on a neural language model with a character-based encoding layer and two BiLSTM layers, it learns contextualised word representations to maintain syntax, semantics and polysemy handling. The characteristics of ELMo[42] are *contextual,* as each word representation is dependent on whole context that word is being used, *Deep,* as it concatenates the word embeddings in all the layers of a deep pre-trained NN, *character based,* so able to handle out of vocabulary terms not came across through training.

4. **Cross-View Training** (CVT)

In ELMo, unsupervised pre-training and task-specific learning take place in two different training stages for two independent models. Cross-View Training [14] integrates them into a single semi-supervised learning technique in which both supervised learning with labelled data and unsupervised learning with unlabelled data on auxiliary tasks improve the representation of a biLSTM encoder. Supervised learning should be used on labelled samples. CVT educates auxiliary predictor units that observe limited views of the input to resemble the expectations of a whole model that sees the entire input on unlabeled samples. The whole model strengthens because the intermediary presentations are shared by both full and auxiliary modules.

5. Universal Language Model Fine-Tuning (ULMFiT)

Why not create a single model that can be used for any categorization task? The major goal of developing a Universal Language Model was to achieve this. As previously said, Language Model is an excellent task for learning the complex characteristics of language and the interdependencies of words in a sentence. To achieve good transfer learning performance on downstream language classification tasks, ULMFiT [16] employs three phases. First step is using *generative pre-trained LM* and then *task-specific fine-tuning* which used two new training techniques for stabilising the fine-tuning process within the layers of the network. *Discriminative Fine-tuning* is inspired by the fact that different layers of LM capture different types of information, and proposes to tune each layer with varied learning rates and *Slanted*

*Triangular Learning Rates*initially linearly increases the learning rate and then gradually decays it, where the increase step is short to make the model converge to a parameter space that fits for the task faster, whereas the decreasing period is long so that can fine-tune better.The final level is to *fine-tune the target task classifier*. Two typical feed-forward layers are added to the pre-trained LM; Concat pooling extracts max-pooling and mean-pooling over the history of hidden states and appends them with the final hidden state; Gradual unfreezing helps to minimise catastrophic forgetting by progressively unfreezing model layers begin from the last. With one epoch, the last layer is unfrozen and fine-tuned. The succeeding lower layer is then unfrozen. This method is repeated until all of the layers have been tuned, after apply a softmax to forecast a target label distribution.

6. Transformer-based Pre-trained Language Models

The principle of transformers [15] made a significant improvement and success over earlier sequence-to-sequence approaches such as Recurrent NN. Consider this: "Abdul Kalam had been an Indian aeronautical scientist who functioned as India's 11th president." For four decades, he worked as a scientist and a science administrator. As a result of his efforts on the development of ballistic missiles, he became renowned as India's Missile Man." All of the underlined words belong to the same entity, even though preserves the correlations across a lengthiest string of words occurring in a sentence is challenging for a machine. Unlike its "sequential" predecessors, Transformers have enabled "parallelization," which has resulted in numerous breakthroughs. Transformer changes source sequences to output sequencing using Attention. Attention is a technique that computes a sequence's representation by correlating different positions in the sequence.

(a) GPT (OpenAI Transformer)

OpenAI GPT [17], short for Generative Pre-training Transformer, extends the unsupervised language model to a much greater scale by training on a massive collection of text based corpora, similar to ELMo. Despite their similarities, GPT and ELMo differ significantly [42]. ELMo employs a shallow/thin mix of individually trained left-to-right and right-to-left multi-layer LSTMs, whereas GPT employs a multi-layer transformer decoder. Contextualized embeddings are used differently in downstream tasks: ELMo sends embeddings into models made for specific activities as added features, whereas for all down-stream tasks the same base model is fine-tuned.

(b) Bidirectional Encoder Representations from Transformers (BERT)

GPT successor, but BERT is bidirectional because it scans text from both the left and right at the same time, capturing both contexts. For example. "She drove to the bank to make a payment.While returning She strolled along the river bank,". As we can see, the meaning of the term "bank" varies greatly between the two statements and is dependent on the both contexts. To capture this, BERT [18] uses bi-directionality via Transformers as multi-layer bidirectional encoders. On a large dataset, using a combination of masked language modelling target and next sentence anticipation, it was pre-trained.

(c) BERT Variants

i. **GPT2 (Zero Shot Learning)**

GPT2 [17] is a general-purpose learner; it has not been particularly trained to execute any of tasks like machine translation, answering questions, summarizing, and generates text output on a level that When generating long paragraphs, indistinguishable from human words, it might become repetitious or meaningless and its ability to do so is a by-product of its general capacity to accurately synthesise the next item in any sequence and can works for downstream tasks with no change in architecture and parameters of pre-trained model.[42] GPT-2 is a ten-fold increase in both the number of parameters and the size of the training dataset over the GPT model. The GPT design uses a transformer model, which replaces earlier recurrence- and convolution-based structures. The model's attention processes allow it to selectively focus on portions of input text that it believes are the most important. This model has a lot more parallelization than earlier RNN/CNN/LSTM-based models, and it outperforms them. It reaches SOTA performance in a zero-shot transfer setting without any task-specific fine-tuning in 7 out of 8 tested language modelling datasets.

ii. **XLNET(Generalized Autoregressive Pre-training for Language Understanding)**

An autoregressive Transformer that have benefits of autoencoding and autoregressive language modelling while trying to avoid their shortcomings. XLNet [19] improves the predicted log likelihood of a sequence based on all possible permutations of the factorization order, rather than adopting a fixed forward or backward factorization sequence as in traditional autoregressive models. The permutation process allows the context for every position comprises tokens from both the left and right sides. Each position is expected to learn to make use of contextual information from all other positions, preserving bidirectional context. XLNet[44] also incorporates the segment recurrence mechanism and relative encoding strategy of Transformer-XL into pre-training, which experimentally increases performance, especially for tasks that involve a lengthier text sequence, as inspired by recent advances in autoregressive language modelling.

iii. **RoBERTa (Robustly Optimized BERT Pre-training Approach)**

[20] Brings some important enhancements to the BERT masked-language modelling training technique. Proposed for improving BERT architecture in order to save time while pre-training. It is built on BERT and changes crucial hyper parameters, removing the job of predicting the next sentence. Training with larger batch sizes, over longer periods of time, and with longer sequences in training data during pre-training improves performance on downstream tasks and avoided utilising a single static mask, dynamically change the masking pattern.

iv. **ALBERT**

A slightly lighter variant of the BERT, ALBERT [21] makes the following three changes: the first two help minimise parameters by factoring the input and hidden layer while the first two propose an embedding matrix for context-independent and context-dependent learning, respectively, and memory requirement, and thus improving the training speed and sharing parameter between among different layers of the method to enhance efficiency and decrease redundancy, while the third one proposes a more daunting training task to replace Instead than using Next Sentence Prediction, ALBERT utilized Sentence Order Prediction in training.

v. **StructBERT**

Taking advantage of word- and sentence-level ordering, enhanced BERT framework well with both word and sentence structural goal to capture language structures in pre-training stage. StructBERT [49] compelled to recreate the proper word and sentence ordering.

(d) Other Types of models

i.Facebook Transcoder Cross Lingual LM Pre-training (XLM)

[22] It's pre-trained transformer for next word prediction goal, a BERT-like masked language modelling objective, and a translation objective.To train AI to comprehend code written in a different language and convert code from one to the other. Many organizations have code written in older programming languages like COBOL, and switching to modern programming languages like Java, C++, or Python requires a significant investment of time, money and resources. In this instance, Transcoder is helpful.They presented three principles for training the model: [46] Cross-programming Language model (XLM) Pre- verifies that a different part of the code representing the same command, independent of programming language, is mapped to the same representation. De-noising Auto Encoding: The model has been trained to anticipate a sequence of words given a distorted version of that sequence. Back Translation: To address the issue that the quality of these translations is often poor because the model has never been trained to translate procedures from one language to another.

ii.DistilBERT

DistilBERT [23] is trained using distillation of Bert that is relatively small, quicker, less expensive, and lightweight transformer model. DistilBERT is a distilled (approx.) version of BERT that maintains 95% of the performance while utilizing half the parameters. In particular, it lacks token-type embeddings, a pool process, and half of BERT layers were retained. DistilBERT[46] employs a process known as distillation to approximate Google's BERT, which entails replacing a huge neural network with a smaller one. After a big neural network has been trained, the entire output distributions of the network can be estimated via a smaller network. This is related to posterior approximation in some way.

iii.SpanBERT

The SpanBERT [24] model is an improvisation of the BERT model by predicting text spans more accurately. Contrasting with BERT, masks arbitrary continuous spans of text rather than arbitrary individual tokens, , then train the model to estimate the completely marked spans using tokens at the start and end of the span border. Instead of Masked Language Modelling and Next Sequence Prediction, the model has been trained on the Span Boundary Objective, which eventually contributes to the loss function.

iv.UniLM

UniLM[25] is a natural language understanding and generating system which is fine-tuned. Three types of language modelling tasks are used to pre-train the model: unidirectional, bidirectional, and sequence-to-sequence prediction. Using a sharing Transformer network and particular self masks to determine which context the prediction relies on, was achieved by unified modelling.

v.ELECTRA

Masking a small fraction of unlabeled inputs and trains the network to reconstruct such actual input is used as pre-training process for some of the language models such as BERT and XLNet. Even thoughit performs effectively, this method is inefficient in terms of data because learning happens from a small number of tokens. A new pre-training strategy introduced to detect the token that was replaced. Rather than masking, some tokens are replaced with acceptable equivalents produced by a tiny language model. The pre-trained discriminator is then applied to each token to determine if it is an actual or a replacement as shown in fig., making the model computationally more effective. [26] A pre-training strategy that involves the training of two transformers: the *generator and the discriminator*; the generator is trained as a masked model that changes tokens in a sequence. The discriminator determines the tokens in the sequences that were altered by the generator.

vi.Text-to Text Transfer Transformer (T5)

By considering every NLP problem as text-to-text, this technique allowed to use the same model along with same objective and training process for any task, by leveraging the prefix attached for each task. The encoder-decoder implementation, which is based on the core Transformer architecture was used in T5 [27], after tokenization, embeddings were found out, passed on through the encoder and then decoder finally produces the output. T5 [47] uses the "Natural Language Decathlon" model, which translates many basic NLP tasks into question-answering in a context. T5 distinguishes task intentions using concise task prefix rather of an explicit QA form, and fine-tunes the model separately for each task. The text-to-text model also enables a simpler transfer learning testing across a range of tasks using the same model.

vii.DeBERTa

[50] Disentangled attention and an improved mask decoder are the two primary enhancements over BERT. DeBERTa comprises of two vectors that indicate a token/word, one for content and one for relative location. Has an upgraded mask decoder, which receives the actual position of the token as well as corresponding data.

viii.**GPT3**

GPT-3 [28] (Generative Pre-trained Transformer 3) is a deep learning-based language model that generates output as like human. It can not only produce text, but also stories, code, poetry, and other types of content. GPT-3 is the successor to GPT-2, and was superior to GPT-2 in terms of size and scope. In fact, when compared to other language models, OpenAI GPT-3's [47] entire version has roughly 175 billion trainable parameters, currently the largest model learned so far. In general, the more parameters a model contains, the more data is needed to train it. The OpenAI GPT-3 model was trained on 45 TB of text data gathered from various sources. Determines which tokens from a known vocabulary will appear next based on the input text.

ix.Bidirectional and AutoRegressive Transformer(BART)

BART [29] a self-supervised auto-encoder utilizes a noise-added given text as input after corrupting a few tokens in the source text or by using any of the appropriate

noising strategies and then reconstructs the actual text by predicting the real substitute of corrupted tokens using an Language Model. The model performs best when employed for Natural Language Generation tasks, although it also works well for comprehension activities.

Table 2: Analysis of Gaps among Classical, Non-contextual, Contextual Language Models

Language Model	Syntax	Semantics	Context	Out of Vocabulary	Word Order	Type	Architecture
One hot encoding and BoW	No	No	No	No	Not considered	Count-based	-
TF, TF-IDF	No	No	No	No			
Word2Vec	Yes	Yes	No	No		Prediction based	Log Bilinear
GloVe	Yes	Yes	No	No		Count-based	
FastText	Yes	Yes	No	Yes		Prediction based	
Context dependent Embeddings of word (Context2Vec, CoVe, ELMo etc.,)	Yes	Yes	Yes	Yes	Preserves sequential order/ position of words		BiLSTM
OPENAI GPT, BERT, XLM etc., transformer based							Transformer

4. Applications

These LMs have been frequently utilized to retrieve features for classification tasks in the early history of ML and AI, particularly in the field of retrieval methods from the given information. Nonetheless, as technology has progressed, continued to evolve, they have become widely used in a variety of fields, including medical, social sciences, healthcare, psychology, text and web mining, law, engineering, and so on. These LMs have been compiled from a variety of

sources. Employed in a variety of text categorization applications and in several fields such as including Information Retrieval and Machine Translation, Sentiment Analysis, Question Answering, Analysis, Recommender Systems, Summarization, Urgency Detection, Market Intelligence, Customer Service Automation, Intent identification, personalized bots, HR, Advertising and trending uses in health-care also. In several fields, adversary attacks, named entity recognition, and defense, and so on.

5. Conclusion

In this article, we've studied a variety of methods for acquiring important data from text input and representing it as vectors for use in standard frameworks. We started with traditional text representation methods, which largely entailed feature engineering, and then moved on to DL-based models. DL approaches have received much interest in recent years, and they're well known for their potential to solve difficulties in computer vision and speech recognition. The efficiency of DL is due to its employment of several layers of nonlinear simple processing units for studying varying levels of characteristics retrieved from data; different layers correspond to different levels of abstraction. Text categorization and natural language processing applications can benefit from deep learning methodologies. Word2vec, GloVe, FastText, Context2Vec, CoVe, and ELMO, as well as context words vectors like Context2Vec, CoVe, and ELMO. Finally, we discussed State-Of-The-Art models based on transformers trained not only on general corpora but on domain-specific transformer-based LMs. There is a lot of scope for further research in this direction to develop effective embeddings for representing natural language objects at various levels. With more and better model designs, we expect this tendency to continue for NLP applications that use reinforcement learning approaches also, such as dialogue systems. We also hope to see further study on multi-modal learning.

References

[1] Hang, Yin, Understanding bag-of-words model: A statistical framework, International Journal of Machine Learning and Cybernetics,

[2] Karen Sparck Jones. 1988. A statistical interpretation of term specificity and its application in retrieval. In Document Retrieval Systems. Taylor Graham Publishing, London, UK, 132–142.

[3] TolgaBolukbasi, Kai-Wei Chang, James Y. Zou, VenkateshSaligrama, and Adam T. Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In Conference on Advances in Neural Information Processing Systems 29, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 4349–4357.

[4] ÁbelElekes, Adrian Englhardt, Schäler, and KlemensBöhm. 2018. Toward meaningful notions of similarity in NLP embedding models. Int. J. Dig. Libr. (04 2018).

[5] YoshuaBengio, RéjeanDucharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. J. Mach. Learn. Res. 3 (Mar. 2003), 1137–1155.

[6] Tomas Mikolov, IlyaSutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Conference on Advances in Neural Information Processing Systems.

[7] MarwaNaili, AnjaHabachaChaibi, and HendaHajjami Ben Ghezala. 2017. Comparative study of word embedding methods in topic segmentation. ProcediaComput. Sci. 112 (2017), 340–349.
[8] Christopher D. Manning, MihaiSurdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. 55–60.
[9] PiotrBojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information.
[10] D.N. Perkins and G. Salomon, Transfer of Learning. Oxford, England: Pergamon, 1992.
[11] Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In Conference on Computational Natural Language Learning.
[12] Bryan McCann, James Bradbury, CaimingXiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors.
[13] Matthew Peters, Mark Neumann, MohitIyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Association for Computational Linguistics, 2227–2237.
[14] Kevin Clark et al. "Semi-Supervised Sequence Modeling with Cross-View Training." EMNLP 2018.
[15] AshishVaswani, Noam Shazeer, NikiParmar, JakobUszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and IlliaPolosukhin. 2017. Attention is all you need. In 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., 6000–6010.
[16] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In 56th Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, 328–339.
[17] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and IlyaSutskever. 2018. Language models are unsupervised multitask learners.
[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805 (2018).
[19] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, RuslanSalakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. CoRR abs/1906.08237 (2019).
[20] Yinhan Liu, MyleOtt, NamanGoyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and VeselinStoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. (2019).
[21] ZhenzhongLan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and RaduSoricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations.

[22] Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining (2019).
[23] Victor Sanh, Lysandre Debut, JulienChaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. A
[24] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. Trans. Assoc. Comput. Ling. 8 (2020), 64–77.
[25] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, JianfengGao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In Conference on Advances in Neural Information Processing Systems. 13063–13075.
[26] Kevin Clark, Minh-ThangLuong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.
[27] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, SharanNarang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. A
[28] Tom B Brown, et al. "Language Models are Few-Shot Learners" NeuriPS 2020.
[29] Mike Lewis, Yinhan Liu, NamanGoyal, MarjanGhazvininejad, Abdelrahman Mohamed, Omer Levy, VesStoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.
[30] UsmanNaseem, Imran Razzak, Shah Khalid Khan, and Mukesh Prasad. 2021. A Comprehensive Survey on Word Representation Models: From Classical to State-of-the-Art Word Representation Language Models.ACM Trans. Asian Low-Resour. Lang. Inf. Process. 20, 5, Article 74 (June 2021).
[31] Gu, Bonwoo, and Yunsick Sung. "Enhanced reinforcement learning method combining one-hot encoding-based vectors for cnn-based alternative high-level decisions." Applied Sciences 11.3 (2021): 1291.
[32] Zhang, Yin, Rong Jin, and Zhi-Hua Zhou. "Understanding bag-of-words model: a statistical framework." International Journal of Machine Learning and Cybernetics 1.1 (2010): 43-52.
[33] Harish, B. S., and M. B. Revanasiddappa. "A comprehensive survey on various feature selection methods to categorize text documents." International Journal of Computer Applications 164.8 (2017): 1-7.
[34] Liu, Qing, et al. "Text features extraction based on TF-IDF associating semantic." 2018 IEEE 4th International Conference on Computer and Communications (ICCC). IEEE, 2018.
[35] Di Gennaro, Giovanni, AmedeoBuonanno, and Francesco ANPalmieri. "Considerations about learning Word2Vec." The Journal of Supercomputing 77.11 (2021): 12320-12335.
[36] Ibrahim, Mohammed, et al. "WOVe: incorporating word order in GloVe word embeddings." arXiv preprint arXiv:2105.08597 (2021).
[37] Joulin, Armand, et al. "Fasttext. zip: Compressing text classification models." arXiv preprint arXiv:1612.03651 (2016).
[38] Malte, Aditya, and Pratik Ratadiya. "Evolution of transfer learning in natural language processing." arXiv preprint arXiv:1910.07370 (2019).

[39] Durrani, Nadir, Hassan Sajjad, and FahimDalvi. "How transfer learning impacts linguistic knowledge in deep NLP models?." arXiv preprint arXiv:2105.15179 (2021).
[40] Mao, Rui, Chenghua Lin, and Frank Guerin. "Word embedding and wordnet based metaphor identification and interpretation." Proceedings of the 56th annual meeting of the association for computational linguistics. Association for Computational Linguistics (ACL), 2018.
[41] Tenney, Ian, et al. "What do you learn from context? probing for sentence structure in contextualized word representations." arXiv preprint arXiv:1905.06316 (2019).
[42] Ethayarajh, Kawin. "How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings." arXiv preprint arXiv:1909.00512 (2019).
[43] Mishra, Swaroop, et al. "Natural instructions: Benchmarking generalization to new tasks from natural language instructions." arXiv e-prints (2021): arXiv-2104.
[44] Topal, M. Onat, Anil Bas, and Imke van Heerden. "Exploring transformers in natural language generation: Gpt, bert, and xlnet." arXiv preprint arXiv:2102.08036 (2021).
[45] Liu, Qi, Matt J. Kusner, and Phil Blunsom. "A survey on contextual embeddings." arXiv preprint arXiv:2003.07278 (2020).
[46] Jiao, Xiaoqi, et al. "LightMBERT: A Simple Yet Effective Method for Multilingual BERT Distillation." arXiv preprint arXiv:2103.06418 (2021).
[47] Koto, Fajri, Jey Han Lau, and Timothy Baldwin. "Discourse probing of pretrained language models." arXiv preprint arXiv:2104.05882 (2021).
[48] https://www.topbots.com/generalized-language-models-cove-elmo/#:~:text=CoVe%20(McCann%20et%20al.,of%20the%20entire%20input%20sentence.
[49] Wang, Wei, et al. "Structbert: Incorporating language structures into pre-training for deep language understanding." *arXiv preprint arXiv:1908.04577* (2019).
[50] He, Pengcheng, et al. "Deberta: Decoding-enhanced bert with disentangled attention." *arXiv preprint arXiv:2006.03654* (2020).