Research Article

A comparative study of Word Embedding Techniques to extract features from Text

[1]Neha Kulkarni, [2]Dr. Ravindra Vaidya, [3]Dr. Manasi Bhate

1Research Scholar,
Dr. Santosh Deshpande
Director,

2HOD, Dept. of MCA,
MES IMCC

3Head, T&P Cell,
MES IMCC

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 23 May 2021

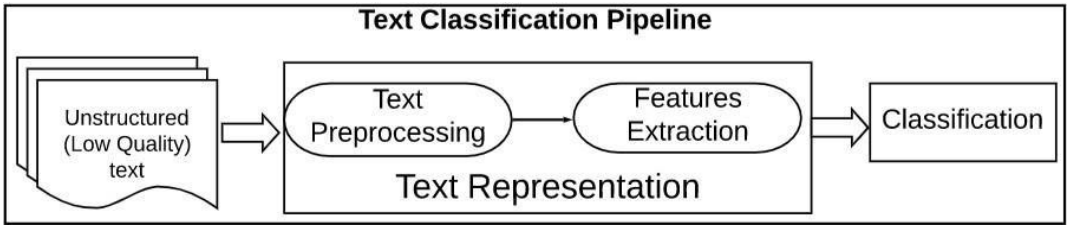
Abstract: Extract information from text into feature vectors is known as word embedding, which is used to represent the meaning of words into vector format. There have been no. of word embedding techniques developed that allow a computer to process natural language and compare the relationships between different words programmatically. In this paper, first, we introduce popular word embedding models and discuss desired properties of word model like similarity analysis, or the testing of words for synonymic relations, is used to compare several of these techniques to see which performs the best.

Keywords: Word embedding, Natural Language Processing, Neural Network, Machine Learning.

1. Introduction:[1][9]

In natural language processing (NLP) there are many algorithms used to achieve the best results, algorithms from Machine Learning (ML), Deep Learning (DL) and many others. The first issue you face in NLP is converting text to numbers that can be used in any algorithm a scientist chooses, but how to convert text to numbers? this is where Word Embedding algorithms come in picture.

Text-based data is increasing at a rapid rate, where the inferiority of the unstructured text is growing rapidly than structured text. Textual data is extremely common in many various domains whether social media, online forums, published articles and online reviews given online where people express their opinions and sentiments to some products or businesses. Text data is a rich source of getting information and gives more opportunity to explore valuable insights which cannot be achieved from quantitative data. The main aim of different NLP methods is to get a human-like understanding of the text. It helps to look at the vast amount of unstructured and low-quality text and find out appropriate insights. Couple with ML, it can formulate different models for the classification of low-quality text to give labels or obtain information based on prior training. Over the years text has been used in various applications such as email filtering, Irony and sarcasm detection document organization, sentiment and opinion mining prediction, hate speech detection, question answering, content mining, biomedical text mining and many more.



2. Word Embedding:[2][8]

Word embedding is a real-valued vector representation of words by embedding both semantic and syntactic meanings obtained from unlabelled large corpus. It is a powerful tool widely used in modern natural language processing (NLP) tasks, including semantic analysis, information retrieval, dependency parsing, question answering and machine translation. Learning a high- quality representation is extremely important for these tasks, yet the question "what is a good word embedding model" remains an open problem. As extensive NLP downstream tasks emerge, the demand for word embedding is growing significantly. As a result, lots of word

embedding methods are proposed while some of them share the same concept.

2.1 Desired Properties of Embedding Models:[2]

Different word embedding models yield different vector representations. There are a few properties that all good representations should aim for.

- Non-conflation
- Robustness Against Lexical Ambiguity
- Demonstration of Multifacetedness
- Reliability
- Good Geometry

3. Word embedding techniques:[7]

Below are the popular and simple word embedding methods to extract features from text are

- Bag of words
- TF-IDF
- Word2vec
- Glove embedding
- Fastest
- ELMO (Embeddings for Language models)

4. Feature Extraction Method:[1][7][4][8]

In this section, we discuss various popularly used feature extraction models. Different features of extraction models are proposed to address the problem of losing syntactic and semantic relationships between words. These methods have been adopted for different NLP related tasks. First, we present some classical models, followed by some famous representation learning models.

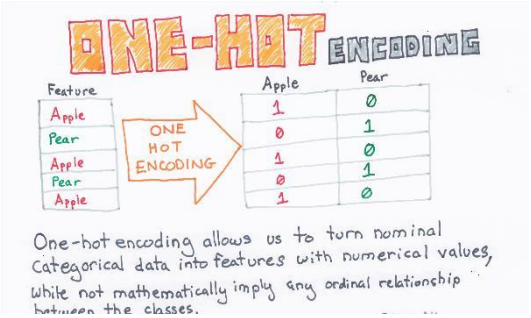
4.1 Classical Models

This section presents some of the classical models which were commonly used in earlier days for the text classification task. Frequency of words is the basis of this kind of words representation methods. In these methods, a text is transformed into a vector form which contains the number of the words appearing in a document.

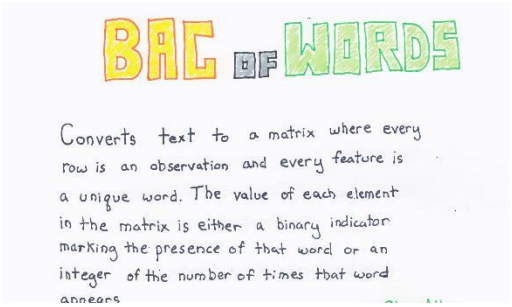
(1) Categorical word representation:

This is the simplest way to represent text. In this method, words are represented by a symbolic representation either "1" or "0".

- One hot encoding: The most straightforward method of text representation is one hot encoding. In one hot encoding, the dimension is the same amount of terms present in the vocabulary. Every term in vocabulary is represented as a binary variable such as 0 or 1, which means each word is made up of zeros and ones.

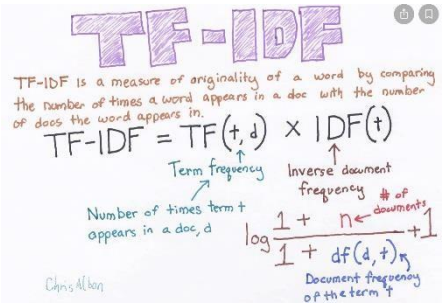


- Bag-of-Words (BoW): BoW is simply an extension of one-hot encoding. It adds up the one-hot representations of words in the sentence. The BOW method is used in many different areas such as NLP, computer vision (CV), and information retrieval (IR) etc.



(2) Weighted Word representation:
Here, we present the common methods for weighted word representations such as Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF). These are associated with categorical word representation methods but rather than only counting; weighted models feature numerical representations based on words frequency.

- Term Frequency (TF): Term frequency (TF), is the straightforward method of text feature extraction. TF calculates how often a word occurs in a document. A word can probably appear many times in large documents as compared to small ones. Hence, TF is computed by dividing the length of the document. In other words, TF of a word is computed by dividing it with the total number of words in the document.



Term Frequency-Inverse Document Frequency (TF-IDF): TF-IDF is presented to cut down the impact of common words such as 'the', 'and' etc. in the corpus. TF means Term frequency which is defined in the above section, and IDF is inverse document frequency which is a technique presented to be used with TF to reduce the effect of common words. IDF assigns a more weight to words with higher or lower frequencies. This combination of TF and IDF method is known as TF-IDF.

4.2 Representation Learning

The limitations of classical feature extraction methods make it use a limited for building a suitable model in ML. Due to this, different models have been presented in the past, which discovers the representations automatically

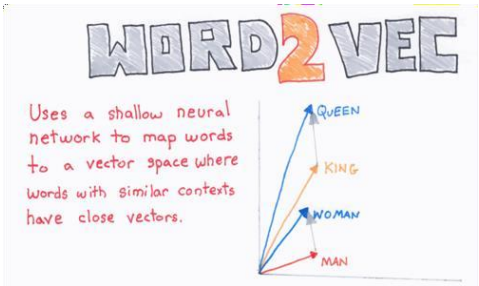
for downstream tasks such as classification. Such methods which discover features itself are called as feature learning or representation learning. In the area of NLP, unsupervised text representation methods like word embeddings have
replaced categorical text representation methods. These word embeddings turned into very efficient representation methods to improve the performance of various downstream tasks due to having a previous knowledge for different ML models. Classical feature learning methods are replaced by these neural network-based methods thanks to their good representation learning capacity. Word embedding is a feature learning method where a word from the vocabulary is mapped to *N* dimensional vector. Many different words embedding algorithms have been presented.

(1) Continuous Words Representation (Non-Contextual Embeddings):

Word Embedding is NLP technique in which text from the corpus is mapped as the vectors. In other words, it is a type of learned representation which allows same meaning words to have the same representation. It is the distributed representation of a text (words and documents) which is a significant breakthrough for better performance for NLP related problems.

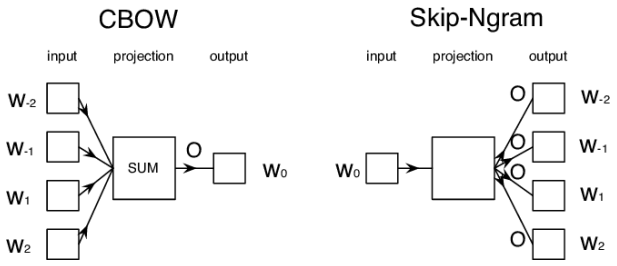
Word2Vec

Word2vec is an efficient analytical model used to transform the raw text into word embeddings. This model is predicated on words with similar semantics present within the same context. this will be modelled by placing a word during a high dimensional vector space then moving words closer supported their probabilities to seem within the same context. Two important methods are used to calculate these vectors
like, Continuous Bag-of-Words model (CBOW) and Skip-Gram model. The advantage of this model is to handle huge volume of documents and provides the optimal results with word vectors.



Continuous Bag of words (CBOW) [5]

Continuous Bag of words (CBOW) gives words prediction of current work based on its context. CBOW communicates with the neighbouring words in the window



Skip-Gram:

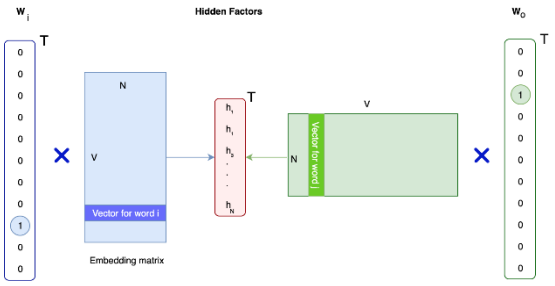
Skip-Gram is the reverse of CBOW model; prediction is given based on the central word after the training of context in skip-gram.

GloVe

The Global Vectors for Word Representation, or GloVe, calculation is an augmentation to the word2vec strategy for efficiently learning word vectors, created by Pennington, et al. at Stanford University. Conventional vector space models expose of words were produced utilizing matrix factorization strategies. GloVe is an approach to extracts both the novel measurements of matrix factorization procedures like LSA with the local context-based learning in word2vec.GloVe constructs an express word-context or word co-occurrence matrix

utilizing statistics over the entire text corpus .The outcome is a learning model is the better embeddings in terms of words.

Word Order Vectors (WOVe) [4]



The next word embedding technique is WOVe , a modification upon GloVe proposed by Cox in 2019 that was able to improve GloVe's
effectiveness in the analogy task by 9.7%. While GloVe does use word-weighting based on those words' distance from the target word when creating the word vector, it does so by generating inclusive matrices. For an inclusive matrix, all words from the target word to the edge of the context window are considered and weighted according to their distance, resulting in a singular vector

FastText [6]

Bojanowski et al. [15] proposed FastText and is based on CBOW. When compared with other algorithms, FastText decreases the training time and maintains the performance. Previously mentioned algorithms assign a distinct representation to every word which introduces a limitation, especially in case of languages with sub-word level information/ OOV

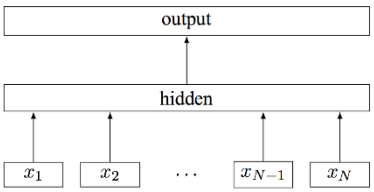
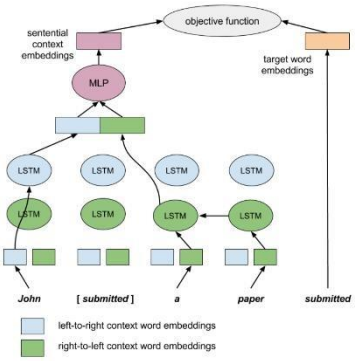


Figure 1: Model architecture of `fastText` for a sentence with N ngram features $x_1, \ldots, x_N$. The features are embedded and averaged to form the hidden variable.

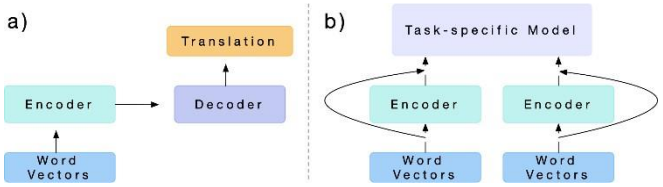
(2) Contextual word representations:

- Generic Context word representation (Context2Vec):

Generic Context word representation (Context2Vec) was proposed by Melamud in 2016 to generate context-dependent word representations. Their model is based on word2Vec's CBOW model but replaces its average word representation within a fixed window with better and powerful Bi-directional LSTM neural network



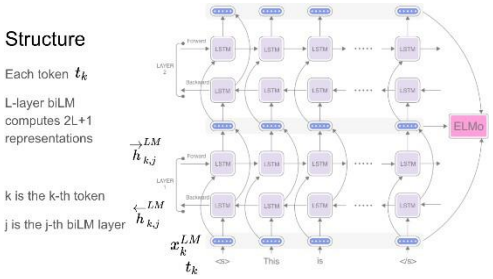
- Contextualized word representations Vectors (CoVe):



McCann presented their model contextualized word representations vectors (CoVe) which is based on context2Vec. They used machine translation to build CoVe instead of
the approach used in Word2Vec (skip-gram or CBOW) or Glove (Matric factorization)

- Embedding from language Models (ELMo):

Peters et al. roposed Embedding from Language Models (ELMo), which gives deep contextual word representations.



5. Analysis of Word Embedding Models: [1][10]

Language Models	Semantics	Syntactical	Context	Out of Vocabulary
1-Hot encoding	[×]	[×]	[×]	[×]
BoW	[×]	[×]	[×]	[×]
TF	[×]	[×]	[×]	[×]
TF-IDF	[×]	[×]	[×]	[×]
Word2Vec	[✓]	[✓]	[×]	[×]
GloVe	[✓]	[✓]	[×]	[×]
FastText	[✓]	[✓]	[×]	[✓]
Context2Vec	[✓]	[✓]	[✓]	[✓]
CoVe	[✓]	[✓]	[✓]	[×]
ELMo	[✓]	[✓]	[✓]	[✓]

6. Comparision of Word Embedding Models [1][3]

Model	Architec t ure	Type	Pros	Cons
One Hot Encoding and BoW	-	Count based	Easy to compute Works with the unknown word Fundamental metric to extract terms	It does not capture the semantics syntactic info. Common words effect on the results Can not capture sentiment of words
TF and TF-IDF	-		Easy to compute Fundamental metric to extract the descriptive terms Because of IDF, common terms do not impact results	It does not capture the semantics syntactic info. Can not capture the sentiment of words
Word2Vec	Log Bilinear	Prediction based	It captures the text semantics syntactic Trained on huge corpus (Pre-trained)	Fails to capture contextual information. It fails to capture OOV words Need huge corpus to learn
GloVe	Log Bilinear	Count based	Enforce vectors in the vector space to identify sub-linear relationships Smaller weight will not affect the training progress for common words pairs such as stop words	It fails to capture contextual information Memory utilization for storage It fails to capture OOV words Need huge corpus to learn (Pre- trained)
FastText	Log Bilinear	Prediction based	Works for rare words Address OOV words issue.	It fails to capture contextual information Memory consumption for storage Compared to GloVe and Word2Vec, it is more costly computationally.
Context2Ve c CoVe ELMo	BiLST M	Prediction based	i) It solves the contextual information issue	Improves performance Computationally is more expensive Require another word embedding for all LSTM and feed-forward layer

7. Conclusion:

The paper has presented multiple techniques used in word embedding and the models and techniques used in those techniques in an attempt to ease the pain of understanding and learning them, it is not considered a full material to learn everything about word embedding techniques but more like an introduction. The main aim of this research work is to analyse the performance of word embeddings algorithm. we have introduced various algorithms that enable us to capture rich information in text data and represent them as vectors for traditional frameworks. We firstly discussed classical methods of text representation. every method has their advantages like a Bag-Of-Words suitable for text classification, TF-IDF is for document classification, WOVe technique for synonyms and if you want semantic relation between words then go with word2vec. We have to choose embedding model depends upon the requirement and corpus.

References Research Papers –

1. Naseem, Usman, et al. "A Comprehensive Survey on Word Representation Models: From Classical to State-Of-The-Art Word Representation Language Models." arXiv preprint arXiv:2010.15036 (2020).
2. Wang, Bin, et al. "Evaluating word embedding models: methods and experimental results." APSIPA transactions on signal and information processing 8 (2019).
3. Janani, R., and S. Vijayarani. "Text Classification: A Comparative Analysis of Word Embedding Algorithms." (2019).
4. Gerth, Tyler. "A Comparison of Word Embedding Techniques for Similarity Analysis." (2021).
5. Almeida, Felipe, and Geraldo Xexéo. "Word embeddings: A survey." arXiv preprint arXiv:1901.09069 (2019).

6. Al-Ansari, Khaled. "Survey on Word Embedding Techniques in Natural Language Processing."

Web References -

1. https://dataaspirant.com/word-embedding-techniques-nlp/
2. https://towardsdatascience.com/document-embedding-techniques-fed3e7a6a25d#d742
3. https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/
[10]https://medium.com/sfu-cspmp/nlp-word-embedding-techniques-for-text-analysis-
4. ec4e91bb886f