A Systematic Literature Review on Document Similarity and Semantic Representation Algorithms

Chenna Keshava B S
CSE Dept, NITK Surathkal
Mangalore, Karnataka
16co108.keshava@nitk.edu.in

Sumukha P K
CSE Dept, NITK Surathkal
Mangalore, Karnataka
sumukhapk46@gmail.com

***Abstract*—This literature review focuses on exploring research papers on document similarity and capturing semantic meaning of documents using Deep Learning models. Document similarity checking algorithms are playing a crucial role in analysing and classifying vast swathes of unlabelled data on the internet. Convolutional Neural Networks have achieved phenomenal success in Computer Vision tasks. Here, we have explored research papers that use neural network models to learn word-vector representations. We also look at ways to generate synthetic datasets in order to apply supervised learning algorithms to NLP tasks. CNNs provide a lot of advantages over Neural Network models in that, they extract the relevant features, irrespective of their location/orientation in the inputs.**

Keywords—Word Embedding, Convolutional Neural Network, Term Frequency Inverse Document Frequency, Clustering, Semantic Interpretation, Information Retrieval.

I. Introduction

In this systematic Literature Review, we have summarised a few research papers pertaining to document similarity and information retrieval. Conventionally, methods like bag-of-words and hand-engineered features for specific tasks were used to achieve better performance in most of the NLP tasks. But the advent of word-vectors revolutionised the research in NLP, as Deep Learning could be applied with a lot of success to almost all of these tasks.

In section 2, we focus on algorithms used for computing similarity between documents. We present a few important points regarding algorithms like tf-idf, cosine distance, learnable string similarities using adaptive duplicate detection.

Machine Learning and Deep Learning algorithms began to be used in NLP since the past two decades. Since there is an extensive amount of literature focussing on Recurrent Neural Networks, especially Long-Short Term Memory Networks (LSTM) in capturing syntactic and semantic meaning of sentences, we have explored the usage of Convolutional Neural Networks (CNNs), which have been successfully used to achieve ground-breaking results in many NLP tasks like search-query retrieval, semantic parsing, sentence modelling, etc. Section 4 has some concluding thoughts on this topics, and some pointers which are worth exploring further. Section 5 contains the references.

II. Distance Measuring algorithms and document similarity.

A. Term frequency Inverse Document Frequency.

- To perform certain operations on a document we need to split the document into manageable parts are retrieve the sub-parts of it. This is called query retrieval and this is quite a major problem on its own. [17] So, if we have a set of documents D, with the user entering a query $q=w_1,w_2,w_3,.....w_n$ for a sequence of words w_i. Then we wish to return a subset D^* of D such that for each d belonging to D^*, we maximise the probability, $P(d|,q,D)$ [18].
- Query retrieval has been done in several methods such as the frequency vector based methods and have provided promising results [19]. Further a weighing scheme called TF-IDF(Term frequency inverse document frequency) is used to solve the query retrieval and weightage of the query problem.
- TF-IDF works by obtaining the ratio of the frequency of the words in a particular document to the frequency of the word in the entire set of documents under consideration. Given a document collection D, a word w, and an individual document d belonging to the set of documents D, we find , $w_d=f_{w,d}*log(|D|/f_{w,D})$,where $f_{w,d}$ equals the number of times w appears in d, |D| is the size of the corpus, and $f_{w,D}$ equals the number of documents in which w appears in D [18].
- Experimentation on TF-IDF had the following consequences:
    - On testing TF-IDF on 1400 documents from the LDCís United Nations Parallel text Corpus, also a naive algorithm that returned the documents with high $f_{w,d}$ was tested along with it.
    - As expected the naive approach was very inaccurate and returned incorrect results for the query.
    - But, the TF-IDF approach returned the documents with the highest frequent use of the query.

  Thus, TF-IDF is quite effective in implementation as well.
- Again, TF-IDF also has its disadvantages. It cannot find relationship between words, cannot find words similar to the query and also plural of the words cannot be found.

B. Clustering

- The process of clustering can be divided into certain steps. Firstly, stop-words are removed. These are the non-descriptive type of words which do not describe the topic whatsoever. Secondly, Porter's suffix-stripping algorithm is deployed [20]. This stems words of the similar origin together making so that they are treated to be the same. Next, words with frequency less than a particular threshold frequency are removed. Thus, top 2000 most frequently used words are used for experimentation. [21]
- Before the process of clustering, a similarity measure must be determined. This measure reflects the degree of closeness of the target objects and also to distinguish between 2 different clusters. Certain measures are discussed below:
    - Metric : A measure 'd' to qualify as metric, d the distance between any two objects 'x' and 'y' in a set, must, be positive, zero iff the two objects of the set are identical, distance from x to y must be same as from y to x

  and the distance must satisfy the triangular inequality.
  - Euclidean distance: It is the standard metric distance for any geometrical problem. The regular formula to find the distance in geometry is applied for the term vectors and their distance is calculated.
  - Cosine similarity: When documents are represented as term vectors, the notation of cosine can be used to find similarity between them. The cosine similarity of a document is defined by the ratio of the dot product of the 2 term vectors and the product of the absolute values of the vectors.
  - Jaccard coefficient : This is the ratio of the intersection and the union of the documents. It can also be written as the ratio of the dot product of the term vectors the square of the difference between the absolute value of them. It ranges between 0 to 1.

- The standard K-means algorithm is used as the clustering algorithm.Here the least squared error is minimised by an iterative partitional clustering method. It takes in a specified number of clusters k and a set of D data objects. Then k data objects are randomly selected as clusters, then the remaining D-k are assigned to these clusters based on the similarity of these to the predefined clusters. Next, new centroids are re-computed for each cluster and in turn all documents are re-assigned based on the new centroids. This particular step is repeated until a fixed solution is reached, that is when all data objects stay in the same cluster after the updation of centroids.
- In practice, the datasets come without predefined categories so this is exactly where clustering can help. In order to make the results of the current investigation comparable to the previous researches, datasets that are commonly used are chosen.

C. ***Learnable string similarity***

- The process of data cleaning and data integration is vital for the current era. The similarity of two datasets are currently done using distance measuring algorithms which have proved quite effective but still lacking certain important characteristics and needs that are vital for these cases.
- Currently the operation on strings for finding the distance between them can be character or vector based methods,each having their own advantages and disadvantages. Character based similarity finding method called the Levenshtein distance is the one which checks the minimum number of insertions, deletions or the substitutions required to get one string from the other.
- The method that is used in the proposed “Learnable string similarity” is to the distance between the two strings’ particular alignment. An alignment is a sequence of pairs of the characters of the 2 strings. Then each pair on being formed, represents an edit operation, substitute, delete or add so that when these operations are performed, one string matches the other.
- One main drawback of this method is that, the calculation of the similarity index is a bit flawed. The probability index of the string pairs that are exactly similar is assigned a value less than one. For the string pairs that

are longer in size the probability index goes on decreasing. This is counter-intuitive. [22]

III. Neural Network Models for NLP

A. Word Embeddings

Word Embeddings is a dense representation of words such that words that have similar meaning have similar representations. [7] The words with similar meanings are close, as determined by methods like cosine or euclidean distance. The vector representations of words are learned in a very similar fashion to that of neural networks. [3] This form of representation is backed by the linguistic theory as detailed by Zellig Harris. It basically hypothesizes that words which have similar context, have similar meanings. [6]

Every word is represented by a real-valued matrix of approximately hundred dimensions. On the other hand, sparse encoding like one-hot encoding will result in matrix that have thousands, if not millions of dimensions. Hence, these dense, distributed low-level vectors for words can be efficiently handled by most of the Neural Network and Numerical Computing libraries.[4]

The feature vector for a word describes the different aspects of the word, so consequently every word can be visualized as a point in the vector space. But the advantage is that, the number of features is much less than the size of the vocabulary, and that gives large profits in real-time production systems like Google Translate, etc. [5]

B. Answer Sentence Selection

We will describe the meaning of some of the technical terms.

Syntactic Parsing is the process of identifying a sentence and assigning a syntactic structure to it. Usually, structures as given by Context-free grammars are used. But Context-Free Grammars do not specify how the parse tree must be generated, syntactic parsing also needs to specify an efficient algorithm to generate correct parse trees for input strings. [10]

Semantic Parsing is the technique of translating a conversational question into a database query, and applying that query to the underlying knowledge base to answer a question [8] Complete understanding of Natural language is not possible only with methods like Vector Space Model(VSM), Dependency Parser, Relation Extraction, etc. In semantic parsing, the natural language query is converted into a formal meaning representation, upon which a machine can act. But the choice of representation depends on our end goal [9].

SEMPRE is a widely used toolkit for training semantic parsers - https://nlp.stanford.edu/software/sempre/. One of the main differences between Semantic Parsing and Machine Translation is that the target semantic representation **is** machine readable in the former, but **not** machine readable in the latter. [9].

Question-Answering(QA) can be dealt with in mainly two ways. First one involves semantically parsing the query and converting it to a database query on a knowledge base. The second method involves many intermediate steps. First the question type is identified, and relevant documents are fetched. Then sentences that might contain the answer are chosen from this set of documents. Then finally the answer is extracted out of these documents. Finally, the relevance of the answer is evaluated based on the semantic matching of the parse trees.

The novel challenges tackled by this paper are, (unlike previous works in this area):

- Choosing an answer sentence, from previously unseen candidate sentences.
- The number of candidate sentences may vary depending on the question.

This model models the problem of choosing an appropriate answer sentence through as a binary classification problem. Every QA is coupled with the judgement(y = Is the answer correct or not?). So the task then becomes classifying if (q, a, y) is a correct tuple or not.

More importantly, this paper uses a unique way to check the relevance of the answer. It 'generates' a question q', from the obtained answer a. Then we perform a semantic similarity on the generated question q' and the original question q. This similarity score is used to judge the relevance of the answer a.
This model can also be extended to tasks like Textual entailment, Paraphrase detection, etc

C. Generating Synthetic Datasets for Supervised Learning in NLP

Using supervised learning for machine reading and understanding is a difficult task, because we do not have a large labelled dataset, and it is hard to develop flexible statistical models that exploit the structure of documents [11]

Researchers have explored creating synthetic narratives/query pairs. This method allows for almost unlimited amount of supervised training data. But such transitions have often been unsuccessful in the history of Computational Linguistics.

This paper aims to build new datasets to facilitate supervised learning in document reading and comprehension by using paraphrase and summary sentences of a document. We can convert it to a context-query-answer triple, and using entity detection and anonymization algorithms, obtain new datasets for machine reading and comprehension.

D. Capturing Semantic Meaning of short-texts using Deep Neural Networks

Before ranking of objects, the objects are mapped from the original words into a feature-vector space where the objects have some type of relationship, semantic relationship(or lexical and syntactic relationship also). [2]
Semantic Relationship is the relationship between words(synonyms, antonyms, homonyms, metanyms, etc), that between phrases and sentences. At the phrase level, the types of relationships are entailment, paraphrases, contradiction, ambiguity, collocation. [1]

Deep Learning models can be trained end-to-end and therefore requires less feature engineering, and easy adaption to new domains in the NLP pipeline. Whenever possible, it is always advantageous to build end-to-end models as Neural networks are very efficient in learning intermediate representations [2]

Distinctiveness of this paper:

- Uses distributional sentence models for mapping input sentences to vectors.
- Apart from the final representation, intermediate ones are also used for similarity scores
- Architecture of the model makes it easy to include additional similarity features into the model
- No manual feature engineering or external resources like are required

- **Prerequisite:** Learn the word embeddings from large, unsupervised corpora, like the blogs, micro blogs, news websites on the internet..

E. CNN for Sentence Classification

This paper applies ConvNets on pre-trained word vectors (as illustrated by Mikolov et al., 2013) for sentence level classification tasks. It performs the classification task with a simple CNN architecture, only using the pre-trained word-vectors as inputs. It's accuracy is comparable to the present best models on multiple benchmarks. This suggests that **pre-trained word vectors are universal feature extractors.**

This work is aesthetically similar to that of Razavian et al. (2014) which proved that the feature vectors for images obtained from a pre-trained deep learning model perform well on a variety of tasks, even the ones very different from the task for which the features were extracted in the first place [13]

F. NLP (almost) from scratch

Many of the recent algorithms in NLP, use task-specific and hand-engineered features in order to achieve impressive results. Although these are useful in practise, they do not help in the broader goals of Natural Language Understanding.

This paper aims to perform well in all of the NLP tasks, by using **minimal** a priori linguistic knowledge. This demonstrates a neural network model that extracts general set of features that can be used to perform different NLP tasks like Part-of-Speech tagging(PoS), Named Entity Recognition(NER) Semantic Role labelling, and Chunking.

This model does not use any externally labelled data for training, and hence can be compared with the other benchmark models also.

G. ConvNet for modelling sentences: [14]

This paper uses a (Dynamic) CNN to semantically model a sentence. The network uses dynamic k-max pooling layers, and it can accept inputs of varying lengths. The network can capture short and long term dependencies, hence has most of the advantages as that of a LSTM (Long Short Term Memory Networks, which is a variant of Recurrent Neural Networks). The network breaks the upper limits of accuracy in small scale binary and multi-class classification and six-way question classification.

Neural Network architectures for capturing semantic meaning of the sentences can be used to extract generic vectors for words and phrases by using the context in which these words appear. Using supervised learning, the vectors for words can be fine-tuned for a specific task.

The architecture uses **dynamic k-max pooling**. K-max pooling is used to sample down different length vectors to a fixed-length vector, before applying it to a fully-connected layer. Dynamic k-max pooling performs the same operation, but the vectors are scaled down commensurate to their input size. [15]

H. Anonymisation of Textual Data [23]

Anonymisation of data is of critical importance in a variety of applications. The performance of different machine learning models can be reliably compared if and only if they have been trained on the same training data set. But since datasets often contain references to sensitive information, due to legal repercussions the data is not being shared today. A similar problem is encountered in medical research where the history of patients' diseases cannot be shared amongst organisations due to the sensitivity of information. Hence, anonymisation of data proves to be a bottleneck in progress of research. Besides, anonymisation is beneficial to NLP models as it helps in **reducing statistical overfitting.**

Anonymisation is a complex problem because the parameters cannot always be clearly defined. The line between what needs to be anonymised and what shouldn't could be different for different corpora and research purposes. For instance, anonymising the brand names in spam/unsolicited email could jeopardise the validity of research in analysing spam emails. [23]

This paper presents a corpus of anonymised text, which serves as a benchmark for evaluating future work in this area. It also provides a pseudonymised version of the same corpus. Pseudonymisation is an anonymization technique where the sensitive references are replaced with other references in the same category. More than anything else, automatic anonymisation techniques could open up vast amounts of personal-text datasets for NLP research. This is especially important in the light of the fact that, current NLP research is hindered due to the lack of structured data, unlike Computer Vision's ImageNet, CIFAR, MNIST, etc.

I. Sentiment Analysis [24]

Sentiment Analysis or Mood Detection is brimming with lot of perspectives today, especially in the aftermath of fake-news campaigns during the elections of USA, November 2016. Even otherwise, this has been a widely used tool by companies to detect the general public opinion regarding their products, marketing, etc.

Besides this use case, sentiment analysis is also used to detect hate-speech, usage of dramatic language in social media websites and consequently, prevent the spread of such dangerous messages.

As of today, most of the sentiment analysis tools can detect the polarity of the document based on the usage of extreme words in the document. But these methods fail to detect the implicit mood of the author in the text, which is not indicated by any particular word/phrase, but is a result of the entire passage. This document gives a comprehensive survey of how the sentiment analysis problem was tackled - the past, present and possible future approaches.

More importantly, classification of documents containing factual data is a very well explored field. But sentiment analysis may play a key role in classifying documents containing subjective opinions of an individual on a particular topic. This is a very important problem that is being tackled today. For instance, analysis of political tweets is a harder problem when compared to that of customer reviews. because in the later, we will be assisted by the ratings provided by the customer. Hence, sentiment analysis for document classification could be worthwhile approach for suture problems.

J. Abstractive Text Summarization

Some of the research approaches aim to perform text summarization first, and then classify the smaller document. This approach needs to be tested for the accuracy aspect of classification. Otherwise, it is bound to be faster than the plain classification algorithms which operate on the entire document.

Historically, text summarization has been dealt with in the extractive manner, where certain important parts of the text are selected and rendered back as the summary. This has the inherent advantage of preserving grammatical correctness and coherency. But it fails to produce medium-large summaries for non-factual data. [25] [27]

On the other hand, abstractive text summarization techniques construct a new summary from scratch, after processing the input document. Needless to say, this is a lot more complex than the extractive text summarization, as it demands real-world knowledge also. [26] [27]

This paper attempts to solve the abstractive text summarization problem by using the neural sequence to sequence model with attention mechanism. In addition, the authors have also explored a reinforcement learning based approach using intra-attention in order to improve the model's performance. [26]

IV. Discussion and findings

The first part of this literature review explored the well-known de-facto algorithms used in the community. The second half explores the importance of semantics in classifying and performing almost all of the NLP tasks. We have explored about 12 papers that we believe could aid us in tackling the problem of Document Similarity in Natural Language Processing.

In section 3, the paper on word embeddings is a quintessential component in techniques for all of the NLP tasks. Hence, exploring it in greater depth helps to gain better insight into most of the recent trends in NLP. The second paper discusses methods for Answer Sentence selection. Since this method can also be extended to other NLP tasks, it serves as a one of the effective strategies worth exploring. More importantly, it solves one of the celebrated problems in NLP - that of, choosing the best answer sentence by considering the context of the question and answer. It also, uses a novel way of *generating a question from the answer*, for evaluating the relevance of the answer. The third paper, establishes methods to generate a rich variety of large-scale synthetic datasets, which have been one of the bottlenecks preventing progress in NLP tasks. Although this is only tangentially related to our chosen topic for Literature Review, it is a significant cornerstone that could potentially improve the results of a model by a significant margin.

The fourth paper in Section 3, is a pioneer in paving way for deep learning methods to explore semantic similarity of words, which was previously achieved only through the tools developed in the 1970s in Computational Linguistics. The fifth paper establishes that the word-vectors are universal feature extractors which are very useful for a wide variety of tasks. The sixth paper demonstrates that generalisation and not using task-specific word vectors enables us to tackle a wide-array of NLP problems using this common features. The last paper uses CNN to model and classify sentences. We feel that this model has a lot of scope to be explored in greater depth in the future.

The last few papers in the previous section present some of the auxiliary features to NLP research like anonymisation and classification using sentiment analysis, which could potentially open up new doors in the future of NLP research.

Serial No.	Paper Title	Authors	Summary

1.	Distributional structure	Harris, Zellig S.	Usage of word embeddings for semantic interpretation
2.	Deep learning for answer sentence selection	Yu, Lei, et al	Two methodologies for answer sentence selection problem
3.	Teaching machines to read and comprehend	Hermann, Karl Moritz, et al	Generating Synthetic Datasets for Supervised Learning in NLP
4.	webdelprofesor.ula.ve/humanidades/azapata/...4/unit_1_semantic_relationships.pdf	Prof. Argenis A. Zapata	Capturing Semantic Meaning of short-texts using Deep Neural Networks
5.	Convolutional neural networks for sentence classification	Hermann, Karl Moritz, et al.	Usage of CNN for learning features for sentence classification
6.	Natural language processing (almost) from scratch	Collobert, Ronan, et al	Minimal usage of a priori knowledge for performing a diverse set of tasks
7.	CNN features off-the-shelf: an astounding baseline for recognition	Razavian, Ali Sharif, et al	Usage of CNN for dynamic modelling of sentences
8.	Introduction to NLP based text anonymization	Medlock, Ben	Advantages of text anonymisation for NLP tasks like classification, POS tagging, chunking
9.	New avenues in opinion mining and sentiment analysis	Cambria, Erik, et al	Application of Sentiment Analysis for sentence classification
10.	Abstractive Text Summarization	Singhal, Soumye, and Arnab Bhattacharya	Abstractive text summarization using neural sequence to sequence learning

V. Conclusion

There is a lot of scope for research in the topics that we have explored. Especially neural network models that use word-vector representations have been able to achieve very good performance in all of the NLP tasks. Also, applying Reinforcement Learning in NLP tasks is a very promising field, as RL complements NLP tasks very well. We hope to continue to explore more research papers in this area.

References

[1] *Recognition Workshops (CVPRW), 2014 IEEE Conference on*. IEEE, 2014.

[2] Prof. Argenis A. Zapata, 2008, Universidad de Los Andes, Escuela de Idiomas Modernos webdelprofesor.ula.ve/humanidades/azapata/...4/unit_1_semantic_relationships.pdf

[3] Severyn, Aliaksei, and Alessandro Moschitti. "Learning to rank short text pairs with convolutional deep neural

networks." *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015

[4] Dr. Jason Brownlee, "What are word embeddings? ", https://machinelearningmastery.com/what-are-word-embeddings/

[5] Goldberg, Yoav. "Neural network methods for natural language processing." *Synthesis Lectures on Human Language Technologies* 10.1 (2017): 1-309.

[6] Bengio, Yoshua, et al. "A neural probabilistic language model." *Journal of machine learning research* 3.Feb (2003): 1137-1155.

[7] Harris, Zellig S. "Distributional structure." *Word* 10.2-3 (1954): 146-162.

[8] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).

[9] Yu, Lei, et al. "Deep learning for answer sentence selection." *arXiv preprint arXiv:1412.1632* (2014).

[10] Bill MacCartney, Introduction to Semantic Parsing, CS224U, Stanford University

[11] Daniel Jurafsky, James H. Martin, Syntactic Parsing, Speech and Language Processing, 2017, https://web.stanford.edu/~jurafsky/slp3/12.pdf

[12] Hermann, Karl Moritz, et al. "Teaching machines to read and comprehend." *Advances in Neural Information Processing Systems*. 2015.

[13] Kim, Yoon. "Convolutional neural networks for sentence classification." *arXiv preprint arXiv:1408.5882* (2014).

[14] Razavian, Ali Sharif, et al. "CNN features off-the-shelf: an astounding baseline for recognition." *Computer Vision and Pattern*

[15] Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom. "A convolutional neural network for modelling sentences." *arXiv preprint arXiv:1404.2188* (2014).

[16] Chen, Yubo, et al. "Event extraction via dynamic multi-pooling convolutional neural networks." *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Vol. 1. 2015.

[17] A. H. M. ter Hofstede H. A. Proper Th. P. van der Weide, "Query formulation as an information retrieval problem."

[18] Juan ramos, "Using TF-IDF to determine word relevance in document queries."

[19] Sivic, Zisserman. "Video Google: a text retrieval approach to object matching in videos"

[20] M. F. Porter. ”An algorithm for suffix stripping."

[21] Anna huang. ”Similarity Measures for Text Document Clustering."

[22]Mikhail Bilenko and Raymond J. Mooney "Adaptive Duplicate Detection Using Learnable String similarity Measures."

[23] Medlock, Ben. "An introduction to NLP-based textual anonymisation." *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC), Genes, Italie*. 2006.

[24] Cambria, Erik, et al. "New avenues in opinion mining and sentiment analysis." *IEEE Intelligent Systems* 28.2 (2013): 15-21.

[25] Gupta, Vishal, and Gurpreet Singh Lehal. "A survey of text summarization extractive techniques." *Journal of emerging technologies in web intelligence* 2.3 (2010): 258-268.

[26] Singhal, Soumye, and Arnab Bhattacharya. "Abstractive Text Summarization."

[27] See, Abigail ., http://www.abigailsee.com/2017/04/16/taming-rnns-for-better-summarization.html