An Overview of Delta Models for Biomedical Information Retrieval

Miloni Mittal
Birla Institute of Technology and Science Pilani
Pilani, India
f20170243@pilani.bits-pilani.ac.in

***Abstract*—Information Retrieval with respect to biomedical literature is a specific area that needs to be focused on due to its peculiar nature of small corpus and large technical vocabulary. The 'Delta' and 'Delta-32-Lex3' models are two deep learning models which have outperformed other state-of-art approaches. They make use of difference of word embedding vectors which is then passed to a neural network. Though the macrostructure of the two models are similar, they differ in minute details which gives an edge to the 'Delta-32-Lex3' model. The models use click logs of the PubMed medical search engine as the dataset. Here, the two models and their aspects are explored so as to understand their structure, results, limitations and any scope for further improvement.**

***Index Terms*—Information Retrieval, Delta models, Biomedical**

I. Introduction

There are millions of articles and journals in the field of biomedicine that showcase the research in the field. PubMed is a platform which helps healthcare workers and researchers access these articles in a more efficient way rather than scavenging through the web among millions of other irrelevant articles. To put into perspective how useful this platform is, PubMed has about 30 million citations in the field as of March 31, 2020 [10] and it caters to about 3 million queries in a day [8]. Owing to the huge amount of text available, there is a need to build an information retrieval system which can cater to the user's information need efficiently by
(i) displaying the search results in minimal run-time and
(ii) more importantly, displaying the most relevant documents at the top of the list.

The major problem faced in building an information retrieval system for this purpose is the availability of a smaller corpus as compared to that of a general web search engine. This leads to difficulty in training the models.

Here, two publications [8], [7] are being analyzed so as to study the methodology followed by them and their limitations. Both propose a deep learning method to retrieve documents and use difference in the word embedding vectors as input. While [8] proposes a 'Delta' model for binary classification of the documents as relevant or irrelevant, [7] proposes a 'Delta-32-Lex3' model for obtaining relevance scores and arranging the documents according to these scores. Both the models aim to divert from the bag-of-words model as used in Okapi BM25 [11] in order to solve issues revolving around bag-of-words model that is, under-specified queries and term mismatches. The approach proposed in [7] is similar to the approach followed in [8] in many aspects but it has some major differences which results in an approximately 10% improvement in the NDCG.20 values in [7]. These differences are pointed out in further sections.

Section II provides references to the research that has been carried out in this field. Section III explains the dataset used in both the publications. Section IV points out the issues that are resolved by the Delta models followed by Section V which explains the structure of the models and compares them parallelly. Section VI compares the results of these two models with other previously proposed ones. Section VII points out some limitations of the delta models and some possible solution. Section VIII concludes the overview with some directives for future research.

II. Past research directions

The models prior to deep learning models used a bag-of-words approach like the Okapi BM25 [11] and Unigram Query Likelihood [5]. These models tend to work on the frequency of the words. This approach posed some limitations (as specified in Section IV) as it failed to identify co-occurrence of terms and the significance of their co-occurrences. Both of these models prove to work better with titles of documents as compared to abstracts and hence the title model is used to compare with other models in Section VI.

Word Mover's distance [3] is a non-parameterised approach which calculates the pairwise distance between vectors of words. More precisely, it finds 'semantically meaningful representations for words from local co-occurrences in sentences' by making use of word embeddings. It finds the similarity between two documents based on these pairwise distances. This method also follows a bag-of-words approach.

Severyn and Moschitti [12] proposed a model which is very similar to the Delta models. It also uses a deep learning approach and makes use of convolutional layers and word2vec but [12] uses a different size of word embeddings. The models mentioned above have been used to compare the results of the delta models as in Section VI.

The authors of [8] indicate that the delta model is very similar to the AlexNet model [2] which was extremely successful in the field of image classification. This model made use of eight layers, which included convolutional layers, max pooling layers and fully connected layers which have been utilized in both the Delta models in different ways.

Numerous neural network models have been surveyed in [6]. It explores different aspects of information retrieval ranging from evaluation metrics, different word embeddings to useful neural networks.

III. The Dataset

The dataset for the study comprised one month of PubMed click logs where the order was specified as ‘Best Match’. For each query that resulted in clicks, the first 20 documents as well as any other clicked document were taken into account. A maximum length limit of 7 and 50 words was imposed on the query and document text (title+abstract) respectively. The document text was either truncated or padded with 0s as needed. Only those queries were taken into account which had at least 21 documents and 3 clicks at minimum, resulting in a corpus of 33,500 queries. For the purpose of training, validating and testing the dataset was split into a 60%, 20% and 20% ratio. Steps were also taken for a fair and unskewed distribution of relevant and non-relevant documents so that the result is not biased. After taking all the above-mentioned measures, the training dataset came to 634,790 query document pairs. Preprocessing steps such as removal of punctuation, tokenization and handling of numbers were also carried out.

In [8], the documents were binary classified as “relevant” if the document was clicked on, to help calculate the precision metrics while evaluating the proposed model. In another scenario, a formula was used to calculate a relevance score and then use it in the NDCG (Normalized Discounted Cumulative Gain) metric.

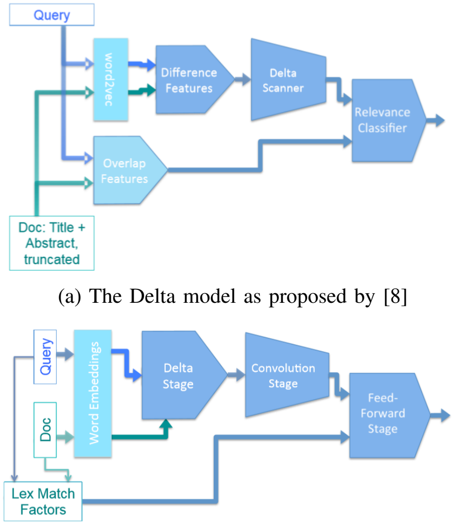
In [7], a measure was designed to assign a relevance score to a query-document pair. This measure was based on the number of click-throughs, availability of document’s full text and number of click-throughs to the document’s full text.

IV. Problems being addressed by the Delta method

The bag-of-words poses two major problems. The first being, “under-specified query problem.” The underlying reason for this is that the bag-of-words model fails to capture the essence of the query and focuses on each word individually. There might be documents present that have a prominent presence of the words in the query but are not actually relevant to the information need of the user. This may lead to retrieval of non-relevant documents (and hence lower precision) and giving them a higher position in the ranks due to the mere presence of the query words in the documents.

The second issue posed is the “term mismatch problem.” This occurs in cases where the query words are not present in the documents. In bag-of-words model, this scenario would lead to ignorance of that term altogether, thus decreasing the number of relevant documents retrieved. The information retrieval system would be more powerful if it has the capability to search for similar meaning words and use it as an alternative.

These issues are catered to by both the delta models by making use of the SkipGram Hierarchical Softmax method which generates corresponding vectors out of words and then that is used to gauge similarity between words.



(a) The Delta model as proposed by [8]

(b) The Delta-32-Lex3 model as proposed by [7]

Fig. 1: The structure of the models

V. The model structure

A. Word Similarity Measurement

These two issues are overcome by the use of SkipGram Hierarchical Softmax method of word2vec [4] producing a word vector of size 300. Using this ensures that words with similar context occupy close spatial positions. The following procedure is used to compute preliminary inputs:

1. Euclidean distance between all possible pairs d_i and q_j is calculated. For each d_i, a corresponding q_i^* is obtained which has the minimum Euclidean distance with d_i. (Where, d_i, q_j and q_i^* stand for the word vectors in the document and query.)

2. In [8], the following three features are computed: d_i-q_i^*, $\|d_i$-$q_i^*\|$, cosine(d_i, q_i^*)

3. In addition to the ones used in [8], [7] uses one more feature: 1-$\|d_i$-$q_i^*\|$/($\|d_i\|$+$\|q_i^*\|$). Therefore, it uses a total of four features.

4. The matrix of these values is used as inputs to the respective neural networks described in Section V-C.

This way the similarity in query and document terms is captured, leading to a better information retrieval system.

B. Overlapping Features

To deal with exact matches between words in query and document, [8] inputs the following features to the relevance classifier stage ‘Overlap Features’ as shown in Fig.1(a): (i) proportion of query and document words in common, (ii) IDF-weighted version of (i), (iii) proportion of query words in the document, and (iv) proportion of query bigrams in the document.

To deal with exact matches, [7] made use of 18 features (‘Lex Match Factors’ as in Fig.1(b)) which included different combinations of Jaccard Similarity, IDF weighted version, BM25 and proportions applied on document text, abstract or title. The 'Delta-32-Lex3' version uses: (i) BM25 on the Document Abstract, (ii) IDF weighted Jaccard Similarity between the Query and the Document Title, and (iii) IDF-weighted proportion of unique Query words in the Document Title. Greedy search was applied to select these three parameters out of the 18 with NDCG.20 as the criterion. 'Delta-32' uses none of these lexical features.

The probable reason why Delta-32-Lex3 model [7] gives a better result in comparison to Delta model [8] is because it selects the best three factors after applying a greedy search approach on 18 factors whereas the latter uses four factors directly without any comparison with other possible features.

C. The Neural Networks Aspect

In [8], the Delta Scanner and Relevance Classifier stage make up the neural networks part of the model like shown in Fig.1(a). The inputs to the Delta Scanner is the matrix produced in Section V-A. This layer consists of vertically stacked convolutional layers, a dropout layer and a global max pooling layer. All of them use the Rectified Linear Unit activation function. The output of this layer and input to next layer is a vector of size 256. The next section of neural networks is the Relevance Classifier. The input to this stage is the vector of size 256 as produced in the previous layer along with the 4 overlap features as described in Section V-B. It consists of a Dropout layer, two feed-forward layers, another Dropout layer, and a sigmoid-based classification layer. The output is a measure of the probability that the document is relevant and the documents are ranked according to this value in the final list of documents. This is how, using sigmoid classification layer, the documents are classified in a binary fashion.

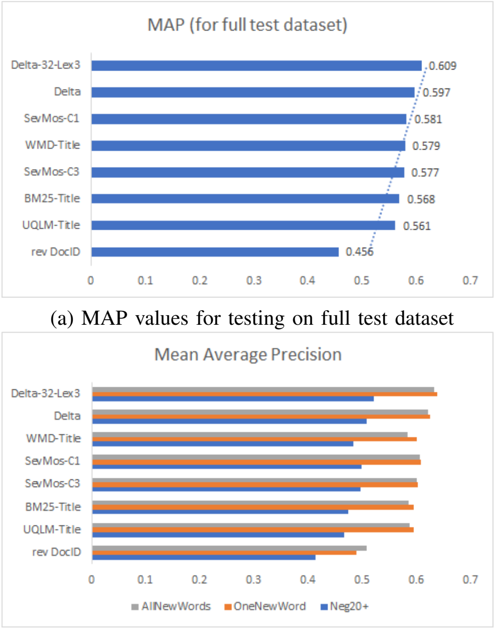
The neural networks stage of [7] consists of a convolutional layer followed by a max pooling layer. The output of this layer is combined with the output of the lexical match features as described in Section V-B to form the input for the next layer which is the feed-forward layer. The activation function used here is the Leaky Rectified Linear Unit. The output is a value which is the relevance score for the document with respect to that query.

Major differences to be noted in these two approaches are:

1. The usage of different neural network layers

2. The different activation functions used (ReLU and Leaky ReLU respectively)

3. The outputs (Binary in the first case and a relevance score in the second case)



(a) MAP values for testing on full test dataset

(b) MAP values for testing of robustness

Fig. 2: Comparison of results of various models

VI. Results of Delta and Delta-32-Lex3 Models

The models were tested on the test dataset and compared with the models prescribed in [3], [12], [11], [5]. The Mean Average Precision values for the proposed methods are summarised through a graph in Fig.2(a). The NDCG.20 metrics are also provided but they cannot be compared due to the different relevance scoring systems used for the click logs in the two publications. The trend line in the graph clearly shows that Delta model with a MAP value of 0.597 [8] outperforms the previously proposed models. Whereas, Delta-32-Lex3 gives an even better performance with MAP value of 0.609. To test the robustness of the models, they were also tested separately on three types of queries:

(i) Neg20+: Queries corresponding to which there are 20 or more non-relevant documents which have query words in their title. (ii) OneNewWord: Queries containing one or more words that were not encountered during training or validation. (iii) AllNewWords: Queries with all words being encountered for the first time.

The results are summarised in Fig.2(b) The Delta models outperform other models in this aspect too. The application of different types of neural networks, different inputs to the last stage and the use of different types of activation functions helps Delta-32-Lex3 model to give better performance metrics.

VII. Limitations

One limitation in the methodology proposed by both [8] and [7] is the disadvantage that comes with the truncation of the document text as described in Section III. The document text is limited to only 50 words and is truncated if this criteria is not satisfied. If a document text length is large, significant amount of information would be lost if the document is truncated. It may so happen that a certain set of useful information occurs in the part of the text that has been truncated thus resulting in loss of relevant information. Moreover, the 'document text' consists of only the abstract and title. It may so happen that certain relevant text are present in the full text and not in abstract or title. These two issues might lead to non-retrieval of relevant documents and hence a lower recall. Resolving this issue of the model might lead to even better results as it would offer full coverage of data.

One limitation that was overcome in [7] was that of binary classification of documents by [8]. This was done by the use of separate kind of neural networks by [7] and elimination of the sigmoid classification layer.

A. *Possible Solution*

The limitation that has been pointed out, can be solved using the following two techniques:

1. The full document text can be summarized by other techniques of natural language processing. Then, this summarized text can be used as input to the Delta models. This would prevent blind truncation of the text and cover the information present in the whole piece of text. Keywords would still be included in the summary which prevents loss of information. [13] proposes a technique to summarise text using long short-term memory and convolutional neural networks. [9], [1] give a review of other summarisation techniques proposed in the past.

2. Another technique that can be used is the assigning of a significance score to all the terms in the vocabulary formed by the corpus. A method can be devised to assign a significance score to words based on how important they are in the biomedical corpus. With this, we can get an understanding of which words, if removed, will not have an impact on the content of the document. A threshold can be decided, and words having significance score less than that threshold can be discarded. This modified corpus can be used as input to the delta model. For example, stop words can be assigned low significance score and medical terminologies can be assigned higher significance scores. By removing words with less importance, words that are more important can be included in the 50-word limit, thus increasing coverage of data.

VIII. Future Scope and Conclusion

One aspect that needs to be tackled specially in the case of biomedical information retrieval is the small size of the dataset in comparison to others such as the web. This implies that similar methods cannot be used in both the types of systems and hence calls for a separate effort towards this direction.

We see that the Delta models use the power of stacked convolutional neural networks to make a more robust model which outperformed other models. This is due to the capability of neural networks to recognize contextual similarity which is very important in classifying a document as relevant or non-relevant. The model is also fast enough to be used as a search engine.

Other aspects that can be explored is the pre-processing of text aspect. In the delta models, there is a lot of computation to be done on the fly due to the 'query-document term comparison' nature of the model. The run-time costs would decrease if a pre-processing method is proposed which complements the Delta models. Also, for more text coverage, techniques like automatic summarisation and keyword extraction can be implemented. One area where both the Delta models aim to improve is deeper semantics. In some cases the models failed to extract the most relevant document at the top of the list. For example, the query 'chronic headache and depression review' failed to extract the most important document 'Psychological Risk Factors in Headache' in the Delta model and in Delta-32-Lex3 it is extracted at rank 5. This shows that some other feature needs to be included and deeper semantics need to be analysed by the model for better results.

References

[1] Mahak Gambhir and Vishal Gupta. "Recent automatic text summarization techniques: a survey". In: *Artificial Intelligence Review* 47.1 (2017), pp. 1–66.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[3] Matt Kusner et al. "From word embeddings to document distances". In: *International conference on machine learning*. 2015, pp. 957–966.

[4] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[5] David RH Miller, Tim Leek, and Richard M Schwartz. "A hidden Markov model information retrieval system". In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. 1999, pp. 214–221.

[6] Bhaskar Mitra and Nick Craswell. "Neural models for information retrieval". In: *arXiv preprint arXiv:1705.01509* (2017).

[7] Sunil Mohan et al. "A fast deep learning model for textual relevance in biomedical information retrieval". In: *Proceedings of the 2018 World Wide Web Conference*. 2018, pp. 77–86.

[8] Sunil Mohan et al. "Deep learning for biomedical information retrieval: Learning textual relevance from click logs". In: *BioNLP 2017*. 2017, pp. 222–231.

[9] Ani Nenkova and Kathleen McKeown. "A survey of text summarization techniques". In: *Mining text data*. Springer, 2012, pp. 43–76.

[10] *PubMed Help*. 2020.

[11] Stephen E Robertson et al. "Okapi at TREC-3". In: *Nist Special Publication Sp* 109 (1995), p. 109.

[12] Aliaksei Severyn and Alessandro Moschitti. "Learning to rank short text pairs with convolutional deep neural networks". In: *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 2015, pp. 373–382.

[13] Shengli Song, Haitao Huang, and Tongxiao Ruan. "Abstractive text summarization using LSTM-CNN based deep learning". In: *Multimedia Tools and Applications* 78.1 (2019), pp. 857–875.