Deep Learning for Biomedical Information Retrieval: Learning Textual Relevance from Click Logs

Sunil Mohan, Nicolas Fiorini, Sun Kim, Zhiyong Lu
National Center for Biotechnology Information
Bethesda, MD 20894, USA
{sunil.mohan, nicolas.fiorini, sun.kim, zhiyong.lu}@nih.gov

Abstract

We describe a Deep Learning approach to modeling the relevance of a document's text to a query, applied to biomedical literature. Instead of mapping each document and query to a common semantic space, we compute a variable-length difference vector between the query and document which is then passed through a deep convolution stage followed by a deep regression network to produce the estimated probability of the document's relevance to the query. Despite the small amount of training data, this approach produces a more robust predictor than computing similarities between semantic vector representations of the query and document, and also results in significant improvements over traditional IR text factors. In the future, we plan to explore its application in improving PubMed search.

1 Introduction

The goal of this research was to explore Deep Learning models for learning textual relevance of documents to simple keyword-style queries, as applied to biomedical literature. We wanted to address two main research questions: (1) Without using a curated thesaurus of synonyms and related terms, or an industry ontology like Medical Subject Headings (MeSH®) (Lu et al., 2009), can a neural network relevance model go beyond measuring the presence of query words in a document, and capture some of the semantics in the rest of the document text? (2) Can a deep learning model demonstrate robust performance despite training on a relatively small amount of labelled data?

We had access to a month of click logs from PubMed®[1], a biomedical literature search engine serving about 3 million queries a day, 20 results per page (Dogan et al., 2009). Most current users of the system are domain experts looking for the most recent papers by an author or search with complex topical boolean query expressions on document aspects. For a small proportion ($\sim 5\%$) of the searches in PubMed, the retrieved articles are sorte by relevance, instead of the default sort order by date. Usage analysis has shown (ibid.) that topic-based queries are a significant part of the search traffic. Such queries often combine two or more entities (e.g. gene and disease), and while users still use short queries, the users are persistent and will frequently reformulate their queries to narrow the search results. So improving the ranking is important to satisfy the needs of PubMed's expanding user base.

Traditional lexical Information Retrieval (IR) factors measure the prominence of query terms in documents treated as bags of words. While such factors like Okapi BM25 (Robertson et al., 1994) and Query Likelihood (Miller et al., 1999) are quite effective, there are several cases where they fail. Two that we wanted to target were: (i) *under-specified query problem*, where even irrelevant documents have prominent presence of the query terms, and relevance requires analysis of the topics and semantics not directly specified in the query, and (ii) the *term mismatch problem* (Furnas et al., 1987), which requires detection of related alternative terms or phrases in the document when the actual query terms are not in the document.

2 Background

Deep Learning models have been applied to various types of text matching problems. Their common goal is to go beyond the lexical bag-of-words

[1] http://ncbi.nlm.nih.gov/pubmed

Proceedings of the BioNLP 2017 workshop, pages 222–231,
Vancouver, Canada, August 4, 2017.

treatment and model text matching as a complex function in a continuous space. An overview of neural retrieval models can be found in (Zhang et al., 2016; Mitra and Craswell, 2017). We review some of this work that motivated our research.

Most text Deep Learning models start with a numeric vector representation of text's lexical units, most commonly terms or words. Ideally these vectors are trained as part of the model, however when training data is limited, many researchers pretrain these word-vectors in an unsupervised manner on a large text corpus, often using one of the `word2vec` models (Mikolov et al., 2013a,b). We used the SkipGram Hierarchical Softmax method to pre-train our word-vectors on Titles and Abstracts from all documents in PubMed.

Word Mover's Distance (WMD) (Kusner et al., 2015) is an (untrained) model for determining the semantic similarity between two texts by computing the pairwise distances between the words' vectors. It leverages the similarity of vectors of semantically related words. When applied to ad hoc IR, it often successfully tackles the term mismatch problem. We compare our model's performance against WMD, and show that the added complexity produces further improvements in ranking.

Many deep learning text similarity and IR models first project the query and each document to vectors to a common latent semantic space. A second stage then determines the 'match' between the query and document vectors. In the relevance model described in (Huang et al., 2013) the last stage is the cosine similarity function, and in follow-up work (Shen et al., 2014) the authors use a convolutional layer as part of the semantic mapping network, and a feed-forward classification network is trained to compute the similarity. Instead of training word embeddings, their document presentation is based on representing each word as a bag of letter tri-grams. Their model is trained on about 30 million labelled query-document pairs extracted from the click logs of a web search engine. The convolution layer is used to capture a word's context and word n-grams. A similar approach is taken in (Gao et al., 2014). The ARC-I semantic similarity model of (Hu et al., 2014) uses a stack of interleaving convolution and max-pooling layers to map a sentence to a semantic vector. They argue that stacking convolutions of width 3 or more allows them to capture richer compositional semantics than the recurrent (Mikolov et al., 2010) or recursive (Socher et al., 2011a,b) approaches. However convolutional architectures do have fixed depths that bound the level of composition. Our use of a vertical stack of convolutional layers without interleaving pooling layers is similar to the successful image recognition models AlexNet (Krizhevsky et al., 2012) and VGGNet (Simonyan and Zisserman, 2015).

Severyn & Moschitti's (2015) model to rank short text pairs is trained on small data ($\sim 50k - 100k$ samples). Word embeddings are pretrained using `word2vec`, a convolutional network maps documents to a semantic vector, followed by a difference matrix and a 3-layer classification network to compute the similarity between the input texts. This is much closer to our final approach, and we compare the performance of our relevance model against this model, but using word-embeddings of size 300 rather than 50 to try to capture richer semantics in biomedical literature.

Another approach to text matching first develops 'local interactions' by comparing all possible combinations of words and word sequences between the two texts. Examples are described in (Hu et al., 2014; Lu and Li, 2013). A recent IR model based on this approach is described in (Guo et al., 2016). Authors argue that the local interaction based approach is better at capturing detail, especially exact query term matches. Our approach simplifies the local-interactions by pairing each document word with a single query word, followed by deep convolutions to attempt to capture some related compositional semantics.

3 The Data

3.1 The Input

We extracted query-document pairs from one month of PubMed click logs where users selected 'Best Match' (relevance) as the retrieval sort order. For each search resulting in a click, the first page of up to 20 documents was recorded. If the clicked document was not on the first page, it was added to this list. The first click on a PubMed search result takes you to a document summary page. Further clicks to the full text of the document were also recorded. Documents that received clicks were labelled as relevant. This binary notion of relevance was used to train our models, and for model evaluation using precision-based ranking metrics. We also experimented with relevance levels, based on a formula hand-tuned to match human-perceived

relevance (see appendix). We report NDCG metrics using these relevance levels.

The queries were restricted to simple text searches, of up to seven words, thus eliminating boolean expressions, author searches and queries mentioning document fields. Log extracts were further restricted to queries with at least 21 documents, and at least 3 clicked documents. These filters reduced the the logs to about 33,500 queries.

These queries were randomly split to 60% training, and 20% each for validation and testing. The number of documents available for each query was quite skewed. Since the metrics we use (described below) give equal weight to each query, we further sub-sampled the training and validation datasets to pick at most 20 of the most relevant documents and an equal number of non-relevant documents. This helped balance out the significance of the queries without reducing the data size too much, and improved the mean per-query metrics of the trained models. The resulting training dataset consisted of 634,790 samples (query-document pairs).

3.2 Pre-processing the Input

We used each document's Title and Abstract to form its text. After some experimentation and evaluation on the validation dataset, we found that limiting this to the first 50 words was optimal. Documents shorter than that were padded with 0's, as were queries shorter than 7 words.

We used a simple tokenizer that split words on space and punctuation, while preserving abbreviations and numeric forms, followed by a conversion to lower-case. All punctuation was dropped, which also resulted in a loss of sentence and some grammatical structure, an area to be explored in the future. Numeric forms were collapsed into 7 classes: Integer, Fraction in (0, 1), Real number, year "19xx", year "20xx", Percentage (number followed by "%"), and dollar amount (number preceded by "$"). Removing stop-words from the query and documents did not improve performance of the models.

We leveraged the large PubMed corpus of about 26 million documents to pre-train the word vectors, using the SkipGram Hierarchical Softmax method of `word2vec` (Mikolov et al., 2013b), with a window size of ± 5, a minimum term-frequency of 101, and a word-vector size of 300. This resulted in a vocabulary of 207,716 words. Rare words were replaced with the generic "UNK" token, which was initialized to $\sim U[-0.25, 0.25]$, as in (Severyn and Moschitti, 2015).

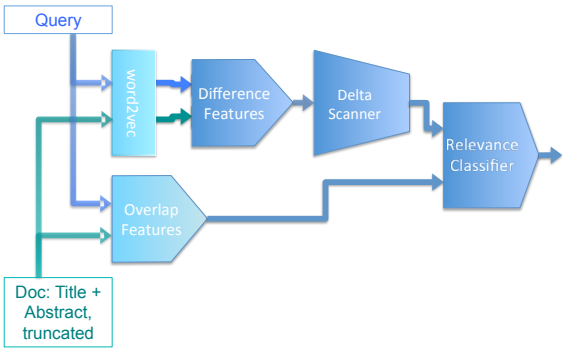

Figure 1: The Delta Relevance Classifier.

4 The Delta Relevance Classifier Model

The components of the Delta Relevance Classifier (figure 1) are described below. Optimal sizes of the various layers were determined by tuning for best accuracy on validation data.

4.1 Note on Convolutional Layers

A convolutional operation (LeCun, 1989) is a series of identical transformations on subsequences of the input obtained by a sliding window on the input. The result is called a feature map, and a convolutional layer will usually involve several feature maps. The width of the input subsequence is called the filter width.

In our application, the input is a sequence of words, each word represented by a real vector of size d. A convolution of filter-width k processes word k-grams. The value of the t-th element of the j-th feature map $\mathbf{c}^j$ is computed as follows:

$$\begin{aligned} \mathbf{c}^j_{[t]} &= \sigma((\mathbf{x} * \mathbf{W}^j)_{[t]} + b_j) \\ \mathbf{x} * \mathbf{W}^j &= \mathbf{x}_{[t-k+1:t]} \cdot \mathbf{W}^j \\ &= \sum_{i=1}^{d} \sum_{j=1}^{m} (x_{i,t-k+j} K_{i,j}) \end{aligned}$$

Here σ is the non-linear activation function, $\mathbf{W}^j \in \mathbb{R}^{d \times k}$, $b_j \in \mathbb{R}$ are the parameters of the j-th feature map, and the input is $\mathbf{x} \in \mathbb{R}^{d \times m}$. In the models in this paper, the feature maps are applied in *full* mode, which effectively pads the input on either side with $k-1$ d-sized 0-vectors, so $x_{i,j} = 0$ for $j < 1$ or $j > m$, and t ranges from 1 to $m + k - 1$.

Applied to text, a convolutional layer of width 3 will extract features from 3-grams. A second con-

	Full Test Data	Neg20+	OneNewWord	AllNewWords
Nbr. of Queries	6,797	2,600	1,823	1,002
Nbr. of Samples	416,509	208,734	90,353	50,827
Prop. of Samples +ive	45.2%	39.5%	48.9%	48.8%
Prop. of Samples -ive	54.8%	60.5%	51.1%	51.2%
+ives without all Query terms in Title	38.9%	13.9%	34.0%	25.2%
-ives with all Query terms in Title	59.1%	83.6%	65.4%	73.4%

Table 1: Test Data and its subsets

volutional layer of the same width stacked above then extracts features from 5-grams, and so on.

4.2 Query-Document Overlap Features

Following (Severyn and Moschitti, 2015), we compute some overlap features to aid relevance detection when dealing with exact matches, and rare words collpased to the 'UNK' token. We use the following overlap features, the first two of which are taken directly from that paper: (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.

4.3 Difference Features Stage

Instead of developing all pairwise local interactions between query and document terms, we capture interactions between pairs of closest terms. This simplifies the model, and since queries are short, we are unlikely to loose any useful interactions. The difference features are computed in two steps (algorithm 1). First, for each word in the document of a query-document pair, the closest query word in absolute vector distance is identified (skipping all "UNK" words in the query and document). We then output the difference vector, along with its length and the cosine angle between the two vectors. With word-vectors of size d and a document of T words, the output of this stage is a real matrix of size $d \times (T + 2)$. We found $T = 50$ produced the best results for the Delta models.

Algorithm 1 Query-Doc Difference Features

Input: Query text $\mathbf{Q}$ and Document text $\mathbf{D}$.
for each word vector w in $\mathbf{D}$ s.t. $w \neq$ UNK **do**:
  Find $w_q = \arg\min(u \in \mathbf{Q}, u \neq \text{UNK}) \|w - u\|$
  Output: $(w - w_q), \|w - w_q\|, \cos(w, w_q)$
end for

4.4 Delta Scanner Stage

The Delta Scanner stage is a vertical stack of three Convolutional layers of 256 filters each, followed by a Dropout layer, and then a Global Max-Pooling layer outputting a fixed-width vector. All feature maps use the *ReLU* (Rectified Linear Unit) activation function.

The input to the Delta Scanner stage is the $d \times (T + 2)$ matrix produced by the Difference Features stage. Documents whose text is fewer than T words are right-padded with 0's, and the Delta Scanner supports a mask input that it uses to ignore the padding. The output of this stage is a vector of size 256, representing the semantic difference between the the query and the document in a query-document pair. The remaining hyper-parameters are: Dropout probability, and the L2-regularization coefficient.

4.5 Relevance Classifier Stage

This is a deep fully connected feed-forward logistic regression stage. The input to the Relevance Classifier stage is the combined vectors output from the Overlap Features and Delta Scanner stages, with a total width of $260 = (4+256)$. This data is fed through the following layers:

i. a Dropout layer,
ii. two feed-forward layers, each of width 260, using the *ReLU* activation function,
iii. another Dropout layer, and
iv. a *sigmoid*-based Classification layer.

The Relevance Classifier's output is an estimate of the probability of the input document's relevance to the query. Documents are ranked in reverse order on this estimated probability.

This stage's hyper-parameters are: Dropout probability (same value used for both Dropout layers), and the L2-regularization coefficient.

4.6 Loss Function and Sample Weighting

The data labels capture a binary sense of relevance, and our models are binary classifiers, so we used the standard binary cross-entropy loss.

In the default mode, the neural network models were trained without any weighting of the training

samples. We trained a second set of models with sample weights derived from the non-binary relevance levels (described above). For each relevance level r, a weight of $\max[1, \log(1+r)]$ was used. This damped the relevance levels, while ensuring that each relevant document received at least the same weight as a non-relevant document.

4.7 Optimization and Implementation Notes

All the neural network models were optimized using Adadelta (Zeiler, 2012), with mini-batches of 256 samples. Mini-batch gradient descent was run for 10 epochs, and the trained values at the end of the epoch producing the best classification accuracy on the Validation dataset were chosen. A greedy search was done in the grid space of the hyper-parameters for the Delta Scanner and Relevance Classifier stages, and the values that produced the best validation accuracy were selected.

5 Experimental Setup

We compare the performance of the relevance models on the following ranking metrics: NDCG at rank 20, Precision at ranks 5, 10 and 20, and Mean Average Precision (MAP). Scoring ties were resolved by sorting on decreasing document-id.

5.1 Methods Compared

We compared the performance of our deep learning model against: BM25; the Unigram Query Likelihood Model (UQLM) with Dirichlet Smoothing (Zhai and Lafferty, 2004); Word Mover's Distance (WMD) that leverages pre-trained word-vectors; and a couple of neural network models based on the architecture described in (Severyn and Moschitti, 2015).

We tested BM25 on the document Title, Abstract and Title + Abstract, and found BM25 on Title to give the best ranking performance, with parameters $k_1 = 2.0, b = 0.75$. Similarly, UQLM applied to the document Title and WMD applied to the document Title after removal of stop-words performed better than the other alternatives.

5.1.1 Severyn-Moschitti Model

We tested four variants of the relevance classifier described in (Severyn and Moschitti, 2015). All versions used the same input data and word-vectors as used for the Delta model. In the basic version, which we will refer to as "SevMos-C1", the query and document were fed into a single-layer Convolutional stage as described in section 4.1, with 256 feature maps and a filter width of 5. This was followed by a Dropout layer and then Global Max-Pooling. The outputs of the query and document convolutions, along with the overlap features described in section 4.2, were fed into a Classifier stage. This stage computed a difference between the query and document features using a difference matrix, and this value along with the other inputs were fed into a deep classification stage identical to that used in the Delta model (section 4.5), sized to match these inputs.

In the "SevMos-C3" variant of this model, we replaced the single-layer convolution stage with a deeper 3-layer stack of convolutions of filter width 3, followed by global max-pooling, just like the Delta model's 'Delta Scanner' stage.

In addition to training the models on unweighted samples, we also trained separate models on relevance-based weighted samples (see section 4.6), which we refer to below as "SevMos-C1 w" and "SevMos-C3 w".

Optimal values for the L2-regularization and Dropout probability hyper-parameters were determined by doing a greedy grid search, as described for the Delta model.

5.2 The Test Data

The test data used to compare performance of the different textual relevance approaches is the held-out 20% split of the data extracted from search logs, as described in section 3.1, without any further sub-sampling. Of the relevant documents ("+ives"), 38.9% did not contain all query terms in the title. Similarly among the non-relevant documents ("-ives"), 59.1% contained all the query terms in the title (see table 1).

In addition to comparing ranking metrics of the different approaches on the test data, we also wanted to explore the main research questions motivating this work: (i) the problems of *under-specified queries* and *term mismatch*, and (ii) *model robustness*. To help answer these questions, we also compare ranking metrics on the following subsets of the test data:

Neg20+: This consists of all queries for which there were at least 20 non-relevant documents that contained all the query words in the title. This helps evaluate performance on under-specified queries.

OneNewWord: The 1,823 test queries which

	NDCG.20	MAP	Prec.5	Prec.10	Prec.20
rev DocID	0.164	0.456	0.344	0.376	0.406
BM25-Title	0.353	0.568	0.592	0.550	0.502
UQLM-Title	0.341	0.561	0.575	0.541	0.500
WMD-Title	0.356	0.579	0.602	0.565	0.516
SevMos-C1	0.345	0.581	0.599	0.569	0.528
SevMos-C3	0.339	0.577	0.597	0.564	0.524
Delta	**0.375**	**0.597**	**0.627**	**0.586**	**0.539**
Delta – WMD	*+5.3%*	*+3.1%*	*+4.2%*	*+3.7%*	*+4.5%*
Delta – SevMos-C1	*+8.7%*	*+2.8%*	*+4.7%*	*+3.0%*	*+2.1%*

Table 2: Ranking metrics on the Full Test Data

contain at least one new word that did not occur in any training or validation queries.

AllNewWords: A smaller subset of queries all of whose words are new: none of the training or validation queries included these words.

The last two subsets will help us evaluate model robustness. The statistics of the test data and its subsets are summarized in table 1.

6 Main Results and Discussion

6.1 Models trained on Un-weighted Samples

Table 2 compares the performance of all the above ranking factors and models on the full test data. The first row shows the metrics obtained by ranking all the documents on reverse order of Document ID. We use this as a score tie-breaker for all the other rankers, so it provides a useful baseline performance of an uninformed ranker.

As also seen in (Shen et al., 2014), BM25 on Title slightly outperforms the Unigram Query Likelihood Model. We have seen other cases where UQLM outperforms BM25. We believe the better performance of BM25 here is partly due to it being a strong factor in the relevance ranking from which these click logs were extracted, thus biasing the click data to some extent.

Word Mover's Distance (WMD-Title) is the first factor in the table that takes non-query words into account, and it does show an improvement over BM25. However WMD relies on the word-vectors obtained by unsupervised training, using a simple Euclidean distance on these vectors as the semantic distance between words. This, and its relatively simple computation, limit how well it performs.

The SevMos-C1 model applies a complex non-linear transformation on the word-vector based text space, in an attempt to better capture comparable semantics of documents. However its NDCG numbers are worse than both WMD and BM25, although its precision numbers, while better than BM25, are about the same as those for WMD. Given that the neural network models in this table were trained on a boolean version of relevance, we expect the main gains to be in the precision-based metrics, which also use a boolean notion of relevance. The lack of improvement in precision metrics over WMD shows that SevMos-C1's non-linear transformations are not doing a better job of capturing query and document semantics.

The SevMos-C3 model learns a more complex non-linear transformation than SevMos-C1, by using a stack of three non-linear convolution layers instead of one in the first part of the model. However its metrics are no better (actually somewhat worse) than SevMos-C1 across the board. So increasing the expressive power of the model did not help. Lack of sufficient training data might be limiting the performance of these models.

The main difference between the Delta model and SevMos-C3 is that the Delta Model starts by computing a difference vector between the Document and Query's word-vector representations. This local interaction vector is inspired by Word Mover's Distance, and in the Delta model we hope to combine the benefits of the WMD and SevMos approaches, while at the same time reducing the complexity of the input space, and thus allowing us to extract more benefit from the small amount of training data. The performance metrics for the Delta model do indeed show sizeable improvements over both WMD and SevMos-C1 (and thus also over BM25 and UQLM). The relative improvements in the metrics are shown in the last two rows of the table[2].

The 'Neg20+' section of the table 3 compares

[2]All cited improvements have been verified to be statistically significant to at least a 99% confidence level using a paired t-test.

	NDCG.20	MAP	Prec.5	Prec.10	Prec.20
Subset: Neg20+					
rev DocID	0.098	0.413	0.310	0.335	0.365
BM25-Title	0.252	0.474	0.490	0.461	0.431
UQLM-Title	0.235	0.466	0.473	0.454	0.428
WMD-Title	0.263	0.483	0.501	0.472	0.441
SevMos-C1	0.277	0.499	0.518	0.492	0.462
SevMos-C3	0.272	0.496	0.519	0.490	0.459
Delta	**0.296**	**0.509**	**0.539**	**0.507**	**0.473**
Delta – WMD	*+12.5%*	*+5.4%*	*+7.6%*	*+7.4%*	*+7.3%*
Delta – SevMos-C1	*+6.9%*	*+2.0%*	*+4.1%*	*+3.0%*	*+2.4%*
Subset: OneNewWord					
rev DocID	0.224	0.490	0.366	0.409	0.443
BM25-Title	0.373	0.595	0.606	0.567	0.520
UQLM-Title	0.368	0.595	0.601	0.567	0.526
WMD-Title	0.362	0.600	0.606	0.578	0.531
SevMos-C1	0.363	0.609	0.614	0.588	0.549
SevMos-C3	0.354	0.603	0.613	0.583	0.547
Delta	**0.402**	**0.625**	**0.644**	**0.606**	**0.559**
Delta – WMD	*+11.0%*	*+4.2%*	*+6.3%*	*+4.8%*	*+5.3%*
Delta – SevMos-C1	*+10.7%*	*+2.6%*	*+4.9%*	*+3.1%*	*+1.8%*
Subset: AllNewWords					
rev DocID	0.230	0.509	0.392	0.439	0.466
BM25-Title	0.352	0.585	0.594	0.564	0.519
UQLM-Title	0.348	0.588	0.593	0.566	0.527
WMD-Title	0.333	0.584	0.582	0.567	0.530
SevMos-C1	0.354	0.607	0.608	0.589	0.554
SevMos-C3	0.340	0.600	0.606	0.578	0.550
Delta	**0.386**	**0.622**	**0.642**	**0.609**	**0.565**
Delta – WMD	*+15.9%*	*+6.5%*	*+10.3%*	*+7.4%*	*+6.6%*
Delta – SevMos-C1	*+9.0%*	*+2.5%*	*+5.6%*	*+3.4%*	*+2.0%*

Table 3: Ranking metrics on selected subsets of the Test Data

	NDCG.20	MAP	Prec.5	Prec.10	Prec.20
Full Test Data					
SevMos-C1 w	0.358	0.586	0.609	0.575	0.531
SevMos-C3 w	0.352	0.582	0.602	0.573	0.528
Delta w	**0.383**	**0.597**	**0.628**	**0.588**	**0.538**
Delta w – SevMos-C1 w	*+7.0%*	*+1.9%*	*+3.1%*	*+2.3%*	*+1.3%*
Delta w – Delta	*+2.1%*	*+0.0%*	*+0.2%*	*+0.3%*	*-0.2%*
Neg20+					
SevMos-C1 w	0.404	0.620	0.635	0.608	0.560
SevMos-C3 w	0.396	0.616	0.635	0.605	0.557
Delta w	**0.427**	**0.630**	**0.653**	**0.617**	**0.564**
Delta w – SevMos-C1 w	*+5.7%*	*+1.6%*	*+2.8%*	*+1.5%*	*+0.7%*
Delta w – Delta	*+0.9%*	*-0.2%*	*-0.9%*	*+0.2%*	*-0.4%*
OneNewWord					
SevMos-C1 w	0.378	0.615	0.628	0.595	0.552
SevMos-C3 w	0.364	0.609	0.619	0.587	0.548
Delta w	**0.408**	**0.624**	**0.644**	**0.606**	**0.558**
Delta w – SevMos-C1 w	*+7.9%*	*+1.5%*	*+2.5%*	*+1.8%*	*+1.1%*
Delta w – Delta	*+1.5%*	*-0.2%*	*+0.0%*	*+0.0%*	*-0.2%*
AllNewWords					
SevMos-C1 w	0.368	0.616	0.629	0.597	0.558
SevMos-C3 w	0.353	0.603	0.609	0.582	0.552
Delta w	**0.389**	**0.621**	**0.642**	**0.607**	**0.562**
Delta w – SevMos-C1 w	*+5.7%*	*+0.8%*	*+2.1%*	*+1.7%*	*+0.7%*
Delta w – Delta	*+0.8%*	*-0.2%*	*+0.0%*	*-0.3%*	*-0.5%*

Table 4: Ranking metrics for Relevance-Weighted models

ranking performance on a subset of the test data that should be harder to rank for factors and models that do not give some consideration to the non-query words in the document. In this data a significant number of non-relevant documents contain all the query words in the title. Comparing these numbers with the previous table shows that there is indeed a significant drop in performance for all the factors and models considered here. However while BM25's NDCG metrics drop by 28.6%, the Delta model's NDCG drops by only 21.1%, with the corresponding drops in MAP being 16.5% and 14.7%, respectively. The Delta model still shows the best metrics on this test set, and its degree of improvement over WMD is bigger, as expected from a more complex model.

Model robustness is tested when queries with words not seen during training (i.e. training and validation datasets) are encountered. This is explored in sections '*OneNewWord*' and '*AllNewWords*' of table 3. Both these sub-tables show a consistently better performance by the Delta model over the other approaches compared here. Interestingly, the improvements in the Delta model's NDCG at 20 metrics over the other approaches are quite sizeable, even though for a simple un-weighted relevance classifier, the primary target was precision and not NDCG.

6.2 Relevance Weighted Models

In this section we explore the performance of the Delta model trained on relevance-weighted samples against the corresponding weighted versions of the neural network models SevMos-C1 and SevMos-C3. These metrics are shown in table 4. A quick comparison with previous tables shows that all the models turn in better NDCG numbers than their un-weighted versions. In particular, the "Delta w" model continues to depict statistically significant better metrics than the other weighted neural network models "SevMos-C1 w" and "SevMos-C3 w".

Comparing the Delta weighted model against the unweighted Delta model, we see that there is a statistically significant improvment in the NDCG metrics for all the Test subsets (at the 99% confidence level). However the precision metrics do not show a significant change. So by weighting the samples we have been able to improve the NDCG without hurting the precision.

7 Concluding Remarks

We have demonstrated a Deep Learning approach for learning textual relevance from a fairly small labelled training dataset. We show that this model is robust and it outperforms both traditional IR factors as well as related shallow (WMD) and deep (SevMos) models based on continuous representations of text, with better results on the under-specified query and term mismatch problems.

While the Delta model is comparable to other local-interaction ranking models, we compute fewer and richer interactions. We believe the fewer interactions captured in the difference vector are sufficient for the shorter queries in our data. As a comparison, the model in (Guo et al., 2016) computes a match histogram based on cosine similarity between all document-query word pairs, and also query-term IDF based weighting. We plan to test this model on our data.

The main advantage to the separate semantic vector approach is that document semantic vectors can be pre-computed. Prediction run-time then primarily depends on the complexity of the similarity computation between these semantic vectors. Local-interaction models, including ours, do not allow this pre-computation, significantly increasing the ranker's run-time cost.

We believe the most promising directions for future research include: modeling deeper semantics (see example in appendix), unsupervised training on data auto-generated from the corpus and fine-tuning with supervised training, improving extraction of non-binary relevance levels and using a pair-wise ranking target. Further investigation is also warranted for incorporating these models into PubMed.

Acknowledgments

This research was supported by the Intramural Research Program of the NIH, National Library of Medicine.

References

Rezarta Islamaj Dogan, G Craig Murray, Aurélie Névéol, and Zhiyong Lu. 2009. Understanding PubMed® user search behavior through log analysis. *Database* 2009:bap018.

G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. 1987. The vocabulary problem in human system communication. *CACM* 30(11):964–971.

Jianfeng Gao, Patrick Pantel, Michael Gamon, Xiaodong He, and Li Deng. 2014. Modeling interestingness with deep neural networks. In *Proceedings of EMNLP 2014*. ACL, Doha, Qatar, pages 2–13.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of CIKM 2016*. ACM, New York, NY, USA, pages 55–64.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in NIPS 27*, pages 2042–2050.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of CIKM 2013*. ACM, New York, NY, USA, pages 2333–2338.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in NIPS 25*, pages 1097–1105.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of The 32nd International Conference on Machine Learning*. JMLR, ICML 2015, pages 957–966.

Yann LeCun. 1989. Generalization and network design strategies. In R. Pfeifer, Z. Schreter, F. Fogelman, and L. Steels, editors, *Connectionism in Perspective*, Elsevier, Zurich, Switzerland.

Zhengdong Lu and Hang Li. 2013. A deep architecture for matching short texts. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in NIPS 26*, pages 1367–1375.

Zhiyong Lu, Won Kim, and W John Wilbur. 2009. Evaluation of query expansion using MeSH in PubMed. *Information retrieval* 12(1):69–80.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of the Workshop at ICLR 2013*.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of INTERSPEECH 2010*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in NIPS 26*, pages 3111–3119.

David R. H. Miller, Tim Leek, and Richard M. Schwartz. 1999. A hidden markov model information retrieval system. In *Proceedings of SIGIR 1999*. ACM, New York, NY, USA, pages 214–221.

Bhaskar Mitra and Nick Craswell. 2017. Neural models for information retrieval. *CoRR* abs/1705.01509. https://arxiv.org/abs/1705.01509.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of TREC 1994*. NIST, Dept. of Commerce, pages 109–126.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of SIGIR 2015*. ACM, New York, NY, USA, pages 373–382.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of CIKM 2014*. ACM, New York, NY, USA, pages 101–110.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *Proceedings of ICLR 2015*.

Richard Socher, Cliff Chiung-Yu Lin, Andrew Ng, and Chris Manning. 2011a. Parsing natural scenes and natural language with recursive neural networks. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of ICML 2011*. ACM, New York, NY, USA, pages 129–136.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011b. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of EMNLP 2011*. ACL, pages 151–161.

Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *CoRR* abs/1212.5701.

Chengxiang Zhai and John Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.* 22(2):179–214.

Y. Zhang, M. M. Rahman, A. Braylan, B. Dang, H. Chang, H. Kim, Q. McNamara, A. Angert, E. Banner, V. Khetan, T. McDonnell, A. T. Nguyen, D. Xu, B. C. Wallace, and M. Lease. 2016. Neural information retrieval: A literature review. *CoRR* abs/1611.06792. http://arxiv.org/abs/1611.06792.

A Document Relevance Levels

Deriving relevance level of a document to a query from observed clicks is still experimental. We use the following formula:

$$\mu \times \text{AbClicks} + (1 - \mu) \times \text{FTClicks}$$
$$+ \frac{1}{\lambda} \times \text{IsDocWithoutFullText} \times \text{AbClicks}$$

with the parameters $\mu = 0.33, \lambda = 15$, where *AbClicks* is the number of observed clicks to the document summary page in PubMed, *FTClicks* is the number of observed clicks to the document's full text, if available, and the value of *IsDocWithoutFullText* is 1 if the full text for that document is not available, and 0 otherwise. The formula attempts to capture the increased notion of relevance if the user accesses the document's full text, without penalizing documents whose full-text is not available. The parameters were hand-tuned to reflect domain experts' relevance judgments.

B Rankings on Some Example Queries

Here are some example queries from the test set showing the titles of the top 3 ranked documents for the Delta weighted model, BM25 and WMD. Relevance levels of the documents are indicated inside parentheses before the titles.

B.1 Query: `cryoglobulinemia`

This word did not occur in training or validation queries. Delta w ranks the most relevant document at the top despite its use of an alternative spelling. BM25 and WMD seem to prefer shorter titles with exact matches. Number of documents in the test dataset: relevant = 27, non-relevant = 26. Top three relevance levels: 39.0, 11.0, 4.0.

As ranked by **Delta w**:

i. (39.0) Diagnostics and treatment of cryoglobulinaemia: it takes two to tango.
ii. (0.0) Clinical features of 30 patients with cryoglobulinemia.
iii. (4.0) The diagnosis and classification of the cryoglobulinemic syndrome.

As ranked by **BM25**:

i. (11.0) Cryoglobulinemia Vasculitis.
ii. (3.0) Cryoglobulinemia (review).
iii. (1.0) Role of CXCL10 in cryoglobulinemia.

As ranked by **WMD**:

i. (11.0) Cryoglobulinemia Vasculitis.
ii. (3.0) Cryoglobulinemia (review).
iii. (3.0) Primary cryoglobulinemia with cutaneous features.

B.2 Query: `oesophageal cancer review`

The word *oesophageal* did not occur in training or validation queries. The word *review* does not occur in the title of all relevant documents. Both Delta w and WMD successfully locate alternative spellings of the word. Number of documents in the test dataset: relevant = 22, non-relevant = 28. Top three relevance levels: 7.0, 4.0, 4.0.

As ranked by **Delta w**:

i. (7.0) Esophageal cancer: Recent advances in screening, targeted therapy, and management.
ii. (3.0) Esophageal cancer: A Review of epidemiology, pathogenesis, staging workup and treatment modalities.
iii. (3.0) Esophageal Cancer Staging.

As ranked by **BM25**:

i. (3.0) Imaging of oesophageal cancer with FDG-PET/CT and MRI.
ii. (0.0) Systematic review and network meta-analysis: neoadjuvant chemoradiotherapy for locoregional esophageal cancer.
iii. (0.0) Serum autoantibodies in the early detection of esophageal cancer: a systematic review.

As ranked by **WMD**:

i. (3.0) Esophageal Cancer Staging.
ii. (0.0) Outcomes in the management of esophageal cancer.
iii. (4.0) Endoscopic Management of Early Esophageal Cancer.

B.3 Query: `chronic headache and depression review`

In this example, both WMD and Delta w are able to leverage word vectors to relate headache to migraine. However both miss the most relevant document, whose title is "Psychological Risk Factors in Headache" (relevance level = 6.0). This example demonstrates the need for deeper semantic modeling. Number of documents in the test dataset: relevant = 23, non-relevant = 18. Top three relevance levels: 6.0, 3.0, 3.0.

As ranked by **Delta w**:

i. (3.0) Migraine and depression: common pathogenetic and therapeutic ground?
ii. (3.0) Migraine and depression comorbidity: antidepressant options.
iii. (3.0) Migraine and depression: bidirectional comorbidities?

As ranked by **BM25**:

i. (3.0) Comprehensive management of headache and depression.
ii. (0.0) Chronic daily headache in children and adolescents.
iii. (0.0) Screening for depression and anxiety disorder in children with headache.

As ranked by **WMD**:

i. (3.0) Comprehensive management of headache and depression.
ii. (3.0) Chronic headaches and the neurobiology of somatization.
iii. (3.0) Migraine and depression: biological aspects.