Denoising Clinical Notes for Medical Literature Retrieval with Convolutional Neural Model

Luca Soldaini
Georgetown University
Washington, DC, USA
luca@ir.cs.georgetown.edu

Andrew Yates
Max Planck Institute for Informatics
Saarbrucken, Germany
ayates@mpi-inf.mpg.de

Nazli Goharian
Georgetown University
Washington, DC, USA
nazli@ir.cs.georgetown.edu

ABSTRACT

The rapid increase of medical literature poses a significant challenge for physicians, who have repeatedly reported to struggle to keep up to date with developments in research. This gap is one of the main challenges in integrating recent advances in clinical research with day-to-day practice. Thus, the need for clinical decision support (CDS) search systems that can retrieve highly relevant medical literature given a clinical note describing a patient has emerged. However, clinical notes are inherently noisy, thus not being fit to be used as queries as-is. In this work, we present a convolutional neural model aimed at improving clinical notes representation, making them suitable for document retrieval. The system is designed to predict, for each clinical note term, its importance in relevant documents. The approach was evaluated on the 2016 TREC CDS dataset, where it achieved a 37% improvement in infNDCG over state-of-the-art query reduction methods and a 27% improvement over the best known method for the task.

CCS CONCEPTS

•Information systems → Query reformulation; •Computing methodologies → Neural networks;

KEYWORDS

medical informatics; convolutional neural networks; query reduction; clinical decision support systems

1 INTRODUCTION & RELATED WORK

The amount of biomedical literature available to health experts has increased dramatically in the last few years. For example, the number of articles in PubMed[1], one of the largest repositories of biomedical literature, grows by approximately 1 million documents each year[2]. This growth is both a blessing and a curse for the medical community: while it enables cutting-edge clinical practices such as evidence-based medicine, it also represents a new set of challenges for health professionals, who often struggle to keep up-to-date with current literature [8].

The interest in clinical decision support (CDS) search systems that could assist physicians in reviewing relevant literature to their clinical practice has been growing in recent years. Such systems are designed to retrieve relevant medical literature given a clinical note describing the conditions of a patient. Since 2014, the TREC CDS shared task[3] has been running as a mean to accelerate research for this application. Several approaches have been proposed to improve CDS search, focusing mostly on query expansion either through domain specific resources (e.g., [7]), pseudo relevance feedback (e.g., [9]), or a combination of the two (e.g., [1, 14, 16]). While CDS TREC 2014 and 2015 relied on fictional clinical descriptions created by health experts, the TREC 2016 dataset [12] provided real clinical notes as search topics. Compared with fictitious clinical descriptions, raw clinical notes present additional challenges for CDS systems, due to "terse language and heavy use of abbreviations and clinical jargon" [12].

In this work, we argue that query reduction techniques that address such challenges ought to be studied, as they improve CDS search by enabling the use of real clinical notes as queries. In particular, we propose a convolutional neural model that is able to predict, for each term in the clinical note, its importance in relevant documents. To do so, it employs several convolutional filters to learn local interactions between terms appearing in clinical notes. Predicted importance is then used to weight terms at retrieval time.

Several domain-agnostic query reduction techniques have been proposed throughout the years. For example, Kumaran and Carvalho [6] introduced a learning to rank approach to find the best sub-query using a series of clarity predictors and similarity measures as features. Bendersky et al. [2] used a supervised method for identifying key concepts in long queries, assigning different weights to concepts extracted from the query. While such techniques are not explicitly designed to handle very long and verbose queries, such as the clinical notes in our dataset, we use them as baseline for the approach presented in this work. In the medical domain, the closest efforts to this work are medical concepts or temporal information extraction from clinical notes (e.g., [3, 15]). However, as noted in [1, 14] and further confirmed in this work, medical concepts alone are not sufficient to express the information need in CDS search, thus justifying our approach.

2 METHODOLOGY

Similar to the work of Kumaran and Carvalho [6] and Bendersky et al. [2], the approach proposed in this paper is designed to predict,

[1] https://www.ncbi.nlm.nih.gov/pubmed/
[2] https://www.nlm.nih.gov/bsd/licensee/baselinestats.html
[3] http://trec-cds.appspot.com/

CIKM'17, November 6–10, 2017, Singapore.

ISBN 978-1-4503-4918-5/17/11…$15.00
DOI: https://doi.org/10.1145/3132847.3133149

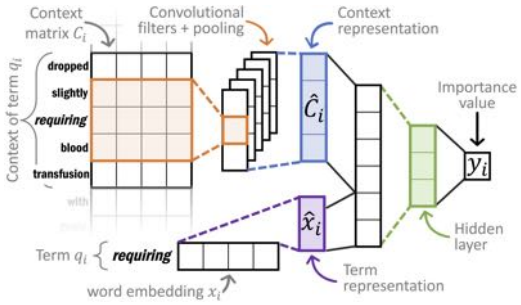

Figure 1: Diagram of the proposed convolutional neural model (CNN). The term being evaluated is *"requiring"*, while the context is *"dropped slightly requiring blood transfusion"*.

for each term in a clinical note, a coefficient that encodes its importance. However, unlike these approaches, we do not use heuristics to select informative query terms, nor we rely on feature engineering to train our supervised method; rather, we use a convolutional neural network (CNN) to directly estimate the importance of each query term by learning from terms in its proximity. Our approach is described in Section 2.1.

To train our model, we use the same strategy employed by pairwise learning to rank methods: given a query, a relevant document for the query, and a non-relevant document for the query, we first use the CNN to determine the weights of terms in the query; then, using these weights, we derive the scores of the two documents; finally, we backpropagate a positive loss if the non-relevant document is scored higher than the relevant document. A more detailed description of the learning strategy is provided in Section 2.2.

2.1 Neural model topology

As previously mentioned, we used a CNN to capture local interactions between terms in clinical notes. On a high level, our system includes several convolution filters of different sizes to exploit interactions between terms in the proximity of each query term; the output of the filters is then reduced to a dense vector, which we refer to as *context representation* $\hat{C}_i$. The context representation of each term is then concatenated with a *term representation vector* $\hat{x}_i$ and used to derive the importance value y_i for each term in the query. A visual overview of the system is presented in Figure 1.

Term representation $\hat{x}_i$: For each query $\mathbf{q} = \{q_1, .., q_n\}$, we first obtain its dense representation $\mathbf{x} = \{x_1, .., x_n\}$. Two source of evidence were used to obtain, for each term q_i, its word embedding x_i: GloVe vectors [10] pre-trained on the common crawl corpus[4] and SkipGram vectors pre-trained on PubMed[5]. We found that concatenating domain-specific with domain-agnostic embeddings yielded the best results; this is consistent with findings in other neural clinical applications [11]. We preserved the case of terms when obtaining word embeddings: this ensures that medical abbreviations, which are often capitalized, are properly captured. In order to reduce the dimensionality, the system learns a task dependent representation of the term feature x_i through a dense layer with ReLU activation function, which we denote as $\hat{x}_i$.

[4]http://commoncrawl.org
[5]https://github.com/cambridgeltl/BioNLP-2016/

Context representation $\hat{C}_i$: For each term q_i in query $\vec{q}$, we define the context of q_i as the c terms preceding q_i and the c terms following it. In other words, the context of q_i consists of the terms appearing in a window of size $2c + 1$ centered in q_i. For each query term q_i, we stack the word embeddings (obtained as described above under "term representation") of the terms in its context to obtain the context matrix $C_i = \{x_{i-c}, .., x_i, .., x_{i+c}\}$. If less than c terms precede q_i or less than c terms follow it, we pad C_i with zeros in order to keep its size consistent with other context matrices.

We chose to define context as the terms appearing in window around each query term, rather than the entire clinical note, as we argue that terms in close proximity to each other contain strong signals that can be used to estimate term relevance, while considering a larger window would add unnecessary noise. Results supporting this observation are presented in Section 4.2. Overall, the approach used to obtain a representation of the context of a term was modeled after the architecture proposed by Severyn and Moschitti [13] to predict similarity between short documents.

To obtain the context representation $\hat{C}_i$, we use convolutional filters of size $k = 2, 3, 4$, and 5, as proposed in [4]. This approach allows to capture local features with different granularities. The convolution layer produces $(c - 2\lfloor k/2 \rfloor)$ features per filter per size (stride size was kept at 1). We indicate the number of filters used for each size as h; we use the same number of filters for each filter size. To reduce dimensionality, we transform each filter using a max pooling layer of size k and stride $\lfloor k/2 \rfloor$ (i.e., from size 2 and stride 1 for $k = 2$ to size 5 and stride 2 for $k = 5$). Finally, after flattening and merging all filters, compact context representation $\hat{C}_{i,c}$ is obtained through a dense layer with ReLU activation function.

We combine term representation $\hat{x}_i$ and context representation $\hat{C}_i$ by concatenation (Figure 1). The resulting layer is first encoded using an intermediate hidden layer with ReLU activation function; then, the predicted importance value y_i for term q_i is obtained by linearly combining the output of the hidden layer, as typically done for regression networks. For simplicity, we will use the notation $y_\theta(\vec{q}) = \{y_1, \ldots, y_n\}^\top$ to indicate the vector of predicted importance values for terms in $\vec{q}$ by the model with weights θ.

2.2 Learning strategy

In order to learn to predict the importance y_i of each query term q_i, we train our model using triples $\langle \vec{q}, \vec{d}_+, \vec{d}_- \rangle$, where $\vec{d}_+$ is a relevant document for the query, and $\vec{d}_-$ is a non-relevant document for the query. In particular, we proceeded as follows: let $Sim(\vec{d}, \vec{q})$ be a function that estimates the similarity of document $\vec{d}$ with query $\vec{q}$. Many similarity functions used in information retrieval (including BM25, which we used in our experiments), are linear with respect to query term coefficients, i.e., they can be written as:

$$Sim(\vec{d}, \vec{q}) = w(\vec{d}, \vec{q}) \cdot \mathbf{1}_n \tag{1}$$

where n is the length of query $\vec{q}$, $w(\vec{d}, \vec{q})$ is a vector of size $1 \times n$ whose elements are the weight of each query term with respect to document $\vec{d}$, and $\mathbf{1}_n$ is a all-ones vector of size $n \times 1$.

In the method we propose, the predicted importance values for terms in $\vec{q}$ are integrated in the similarity function as follows:

$$Sim(\vec{d}, \vec{q}) = w(\vec{d}, \vec{q}) \cdot y_\theta(\vec{q}) \tag{2}$$

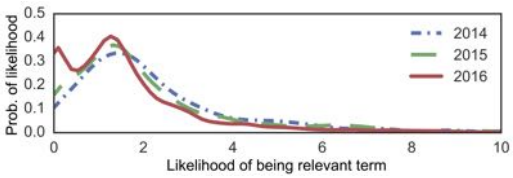

Figure 2: Probability density function of likelihood of being relevant for query terms in the 2014 (blue dashes & dots), 2015 (green dashes), and 2016 (solid red) datasets. Since the distributions are comparable, we augment the training set with the 2014 and 2015 datasets.

Leveraging this notation, we can finally define a pairwise max margin loss function with respect to the training triple $\langle \vec{q}, \vec{d}_+, \vec{d}_- \rangle$ and model weights θ:

$$\mathcal{L}_\theta(\vec{q}, \vec{d}_+, \vec{d}_-) = \max\left(0, 1 - w(\vec{d}_-, \vec{q}) y_\theta(\vec{q}) + w(\vec{d}_+, \vec{q}) y_\theta(\vec{q})\right) \quad (3)$$

We combine the loss function defined in Equation 3 with a regularizing function designed to prevent the model from assigning negative importance to query terms:

$$O(\vec{q}, \vec{d}_+, \vec{d}_-; \theta) = \mathcal{L}_\theta(\vec{q}, \vec{d}_+, \vec{d}_-) + \sum_{y_i \in y_\theta(\vec{q})} \min(0, y_i)^2 \quad (4)$$

We train the proposed model by minimizing this objective function.

3 EXPERIMENTAL SETUP

3.1 Dataset

We studied the effectiveness of the proposed method on the 2016 TREC CDS dataset [12]. It is comprised of 30 topics (each containing a clinical note), 1.25 million articles from the open access subset of PubMed Central[6], and 28,349 documents whose relevancy to topics have been assessed. On average, clinical notes in this dataset have a length of 184 terms and a median of 188; for each note, an average of 182 documents were found to be relevant (median: 119).

Because of the limited amount of training data proved by the 2016 TREC CDS dataset, we expanded the training set using fictitious clinical descriptions from previous years' collections. While descriptions are substantially shorter than actual clinical notes (average length: 81 terms), the distribution of query terms that are likely to appear in relevant documents is sufficiently similar to the one of query terms in the clinical notes dataset (Figure 2); the likelihood of a query term being relevant was defined as the probability of appearing in relevant documents for a query over the probability of appearing in non-relevant documents for the query.

3.2 Model training

We partition the 2016 dataset in training, development, and test sets. The system was evaluated under three-fold cross validation by rotating the subsets. For all three runs, the training set was always expanded using the 2014 and 2015 TREC CDS datasets.

Optimal model topology was determined through empirical evaluation on the development set. The size of the context and term representation layers was set to 128, while the size of the hidden layer was set to 64. To prevent over-fitting, outputs of all layers (except the last one) were regularized using batch normalization; batch size was set to 32. A 30% dropout was also applied at training time to the input of all layers denoted by a dashed line in Figure 1. As illustrated in Section 4.2, we experimented with several filter sizes k; the number of filters per size was set to $h = 256$.

The model was trained using the Adagrad optimizer [5]. Each fold was trained until no improvement in infNDCG was achieved on the development set for 30 epochs (at the end of training, the model was rolled back to the last iteration with improvement).

[6]https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/

		Query reduction approach	TREC CDS 2016 infNDCG	TREC CDS 2016 P@10
baselines	*i*	No query reduction	0.1138	0.1967
	ii	*idf* filter	0.1312	0.2067
	iii	UMLS medical concepts filter	0.1580	0.2400
	iv	Wikipedia medical concepts filter	0.1670	0.2300
st. of the art	*v*	QQP [6]	0.1312	0.2133
	vi	Health-QQP [14]	0.1520	0.2433
	vii	PCW [2]	0.1833	0.2900
	viii	NKU [18] *(best at CDS TREC 2016)*	0.1978	0.2900
	ix	**CNN *(this work)***	**0.2518**	**0.3167**

Table 1: Performance of the proposed approach (*ix*), several baselines (*i* to *iv*), and state of the art methods (*v*-*viii*) on the TREC CDS 2016 dataset. The proposed method (*ix*) shows statistically significant improvements over all other methods (paired Student *t*-test, $p < 0.05$).

4 RESULTS

4.1 Retrieving medical literature

Performance was measured using the two main metrics of 2016 TREC CDS track: inferred NDCG [17] (primary metric) and precision at 10 retrieved results (P@10). The proposed method was compared with several well-known query reduction techniques, as well as the best approach from TREC CDS 2016. In detail, we compared the proposed method with the following approaches (reported in Table 1; *i* to *iv* are baselines, while *v* to *viii* are state-of-the-art techniques):

(i) No query reduction: we left the clinical note as-is, except removing numbers, stop words, and units of measurement.
(ii) Idf filter: we removed terms whose *idf* is less than 1 (term appears in more than 10% of the documents) and more than 5.5 (term appears in less than 3 documents in the collection); values were determined through manual tuning on the development set.
(iii) Query reduction via UMLS concept mapping: we mapped expressions in the query to concepts in UMLS medical thesaurus[7] using QuickUMLS [15]; terms that are not in UMLS were removed.
(iv) Query reduction using Wikipedia: for each term q_i in a clinical note, we estimated its probability of being a medical term by calculating its likelihood of appearing in health pages on Wikipedia.
(v) Query quality predictors (QQP): we implemented the method proposed in [6] to reduce clinical notes. This method uses quality predictors as features to learn to rank sub-queries of clinical notes.
(vi) Medical QQP: we tested a variant of QQP introduced in [14]; this formulation is better tailored to this application, as it considers health-oriented features alongside original predictors.

[7]The Unified Medical Language System or UMLS is a thesaurus for medical terminology maintained by the U.S. National libary of Medicine.

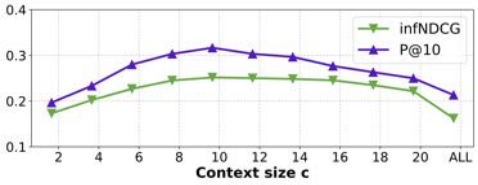

Figure 3: Impact of context size c on method performances.

(vii) Parameterized concept weighting (PCW): we implemented the supervised model introduced by Bendersky et al. [2] to learn weights of concepts in the query. This model uses statistical features (e.g., term and document frequency in target collection) to learn the importance weight of three concept types: unigrams, bigrams phrases, and proximity bigrams. We expanded the set of concept types with medical concepts extracted with QuickUMLS [15], and the set of features with term and document frequencies of candidate concepts in several medical collections.

(viii) NKU team: we compared our system with the work of Zhang and Liu [18], which obtained the best performance on clinical notes at TREC 2016. This method combines concept extraction, query expansion using the MeSH[8], and pseudo relevance feedback.

As shown in Table 1, the proposed CNN (Table 1, line *ix*) outperformed all baselines and state-of-the-art methods. The difference between the proposed CNN and the other methods' performance is more prominent in terms of inferred NDCG, as we observed an improvement of 121% over the unmodified clinical note (line *i*), 37% over the best general domain query reduction (PCW, line *vii*), and 27% over the best system proposed for this task (NKU, line *viii*).

The proposed CNN showed a less pronounced improvement over state of the art methods in terms of P@10; nevertheless, it outperforms all state of the art methods by at least 9% (line *viii*) and up to 30% (line *v*). We attribute this outcome to the fact that the proposed method was trained to maximize the difference in scores between relevant and non-relevant documents; thus, it suffers in precision-oriented metrics with early cutoff, such as P@10.

Finally, we observed that approaches that explicitly take advantage of domain specific resources, such as medical concept extraction using UMLS (*iii*) and Medical QQP (*vi*) outperform methods that do not leverage such resources (*iv* and *v*). This confirms the finding of [1] and [14].

4.2 Choice of hyperparameters

We studied the impact of the hyperparameters detailed in section 2.1 on performance of the proposed method. In detail, we conducted two experiments: we evaluated the impact of context size c on infNDCG and P@10 (Figure 3), and we performed an ablation study to quantify the impact of convolutional filter sizes (Table 2).

We experimented with context sizes ranging from $c = 2$ (that is, considering two terms before and two terms after each query term) to using the entire clinical note as context (c = ALL). As shown in Figure 3, the best performance is obtained when $c = 10$. While the performance of the system is not affected by small deviations from the optimal value, choosing a context that is too small ($c \leqslant 4$) or too large ($c \geqslant 15$) notably reduced its effectiveness. In particular, we note that the model that uses the entire clinical note as context performed worse than any other context size c in terms of infNDCG, supporting our decision to limit the context size.

[8]https://www.nlm.nih.gov/mesh/

Size(s) of convolutional filters used	infNDCG	P@10
$k = 2$	0.2342	0.2867
$k = 2, 3$	0.2435	0.3033
$k = 2, 3, 4$	0.2498	0.3100
k = 2, 3, 4, 5	**0.2518**	**0.3167**

Table 2: Ablation study on the size of convolutional filters.

Finally, we evaluated the impact of the convolutional filters size k using an ablation study. The results presented in Table 2 suggest that using multiple values for k has positive impact on capturing local features, as each filter size learn specific aspects of term interaction in the context. However, we note that the improvement in performance got smaller as larger filters were introduced in the model.

5 CONCLUSIONS

We proposed a convolutional neural model to reduce noise in clinical notes to be used for medical literature retrieval. For each term in a clinical note, the proposed approach takes advantage of the context surrounding the term to predict its importance. The proposed approach was evaluated on the TREC CDS 2016 dataset, and compared several query reduction baselines, as well as state of the art methods, outperforming them all.

REFERENCES

[1] Saeid Balaneshin-kordan and Alexander Kotov. 2016. Optimization method for weighting explicit and latent concepts in clinical decision support queries. In *ICTIR*.

[2] Michael Bendersky, Donald Metzler, and W Bruce Croft. 2011. Parameterized concept weighting in verbose queries. In *SIGIR*. ACM.

[3] Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. Semeval-2016 task 12: Clinical tempeval. *SemEval* (2016).

[4] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv:1408.5882* (2014).

[5] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980* (2014).

[6] Giridhar Kumaran and Vitor R Carvalho. 2009. Reducing long queries using query quality predictors. In *SIGIR*.

[7] André Mourao, Flávio Martins, and Joao Magalhaes. 2014. NovaSearch at TREC 2014 clinical decision support track. In *TREC*.

[8] Eitan Naveh, Tal Katz-Navon, and Zvi Stern. 2015. Resident physicians' clinical training and error rate: the roles of autonomy, consultation, and familiarity with the literature. *Advances in Health Sciences Education* 1 (2015), 59–71.

[9] Heung-Seon Oh and Yuchul Jung. 2015. Cluster-based query expansion using external collections in medical information retrieval. *Journal of Biomedical Informatics* (2015).

[10] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *EMNLP*.

[11] Kirk Roberts. 2016. Assessing the Corpus Size vs Similarity Trade-off for Word Embeddings in Clinical NLP. In *ClinicalNLP workshop at COLING 2016*.

[12] Kirk Roberts, Dina Demner-Fushman, Ellen M Voorhees, and William R Hersh. 2017. Overview of the TREC 2016 Clinical Decision Support Track.. In *TREC*.

[13] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In *SIGIR*.

[14] Luca Soldaini, Arman Cohan, Andrew Yates, Nazli Goharian, and Ophir Frieder. 2015. Retrieving medical literature for clinical decision support. In *ECIR*.

[15] Luca Soldaini and Nazli Goharian. 2016. QuickUMLS: a fast, unsupervised approach for medical concept extraction. In *MedIR Workshop at SIGIR*.

[16] Luca Soldaini, Andrew Yates, and Nazli Goharian. 2017. Learning to Reformulate Long Queries for Clinical Decision Support. *JASIST* (2017). DOI:http://dx.doi.org/10.1002/asi.23924

[17] Emine Yilmaz, Evangelos Kanoulas, and Javed A Aslam. 2008. A simple and efficient sampling method for estimating AP and NDCG. In *SIGIR*.

[18] Hualong Zhang and Liting Liu. 2017. NKU at TREC 2016: Clinical Decision Support Track. (2017).