Phrase2VecGLM: Neural generalized language model–based semantic tagging for complex query reformulation in medical IR

Manirupa Das†, Eric Fosler-Lussier†, Simon Lin‡, Soheil Moosavinasab‡,
David Chen‡, Steve Rust‡, Yungui Huang‡ & Rajiv Ramnath†
The Ohio State University† & Nationwide Children’s Hospital‡
{das.65, fosler.1, ramnath.6}@osu.edu

Abstract

In fact-based information retrieval, state-of-the-art performance is traditionally achieved by knowledge graphs driven by knowledge bases, as they can represent facts about and capture relationships between entities very well. However, in domains such as medical information retrieval, where addressing specific information needs of complex queries may require understanding query intent by capturing novel associations between potentially latent concepts, these systems can fall short. In this work, we develop a novel, completely unsupervised, neural language model–based ranking approach for semantic tagging of documents, using the document to be tagged as a query into the model to retrieve candidate phrases from top–ranked related documents, thus associating every document with related concepts extracted from the text. For this we extend the word embedding–based generalized language model (GLM) due to (Ganguly et al., 2015), to employ phrasal embeddings, and use the semantic tags thus obtained for downstream query expansion, both directly and in feedback loop settings. Our method, evaluated using the TREC 2016 clinical decision support challenge dataset, shows statistically significant improvement not only over various baselines that use standard MeSH terms and UMLS concepts for query expansion, but also over baselines using human expert–assigned concept tags for the queries, on top of a standard Okapi BM25–based document retrieval system.

1 Introduction

Existing state-of-the-art information retrieval (IR) systems such as knowledge graphs (Su et al., 2015; Sun et al., 2015), or information extraction techniques centered around entity relationships (Ritter et al., 2013), that often rely on some form of weak supervision from ontological or knowledgebase (KB) sources, tend to perform quite reliably on fact-based information retrieval and factoid question answering tasks. However, such systems may be limited in their ability to address the complex information needs of specific types of queries (Roberts et al., 2016; Diekema et al., 2003) in domains such as clinical decision support (Luo et al., 2008) or guided product search (Teo et al., 2016; McAuley and Yang, 2016), due to: 1) complex and subjective, or lengthy nature of the query containing multiple topics, 2) vocabulary mismatch between the query expression and knowledge representations in the document collection, and 3) lack of sufficiently complete knowledge bases of “related concepts”, covering all possible relations between candidate concepts that may exist in a collection, essential for effectively addressing these types of queries (Hendrickx et al., 2009).

We hypothesize, that similar to human experts who can determine the aboutness of an unseen document by recalling meaningful concepts gleaned from similar past experiences via shared contexts, a completely unsupervised machine learning model could be trained to associate documents within a large collection with meaningful concepts discovered by fully leveraging shared contexts within and between documents, thus surfacing “related” concepts specific to the current context (Lin and Pantel, 2002; Halpin et al., 2007; Xu et al., 2014; Kholghi et al., 2015a; Turney and Pantel, 2010; Pantel et al., 2007; Bhagat and Hovy, 2013; Hendrickx et al., 2009). As a trivial example, ordinarily unrelated concepts (noun phrases,
in this work) such as “scarlet macaw” and “raccoon” occurring in separate documents $d_1$ and $d_2$ may become related by a novel context such as “exotic pets” that may occur as terms in a query or as a phrase in a document $d_p$ which could be related to both $d_1$ and $d_2$. If by some means, documents $d_1$ and $d_2$ were semantically tagged with the phrase “exotic pets” via $d_p$, those documents would surface in the event of such a query (Hendrickx et al., 2009; Bhagat and Ravichandran, 2008). This could thus help to better close the vocabulary gap between potential user queries and the documents. To our knowledge, ours is the first work that employs word and phrase-level embeddings for local context analysis in a pseudo-relevance feedback setting (Xu and Croft, 2000), using a language model-based document ranking framework, to semantically tag documents with appropriate concepts for use in downstream retrieval tasks (Kholghi et al., 2015a; De Vine et al., 2014; Sordoni et al., 2014; Zhang et al., 2016; Zuccon et al., 2015; Tuarob et al., 2013).

The main contributions of our work, are as follows: 1) We present a novel use for a neural language modeling approach that leverages shared context between documents within a collection via phrase-based embeddings (1, 2, and 3-grams), finding the right trade-off between the local context within each term versus its global context within the collection, incorporating a local context analysis-based pseudo-relevance feedback mechanism (Xu and Croft, 2000) for concept extraction. 2) Our method is fully unsupervised, i.e. it includes no outside sources of knowledge in the training, leveraging instead the shared contexts within the document collection itself, via word and phrasal embeddings, mimicking a human that potentially reads through the documents in the collection and uses the seen information to make relevant concept tag judgments on unseen documents. 3) Our method presents a black-box approach for tagging any corpus of documents with meaningful concepts, treating it as a closed system. Thus the concept associations can be pre-computed offline or periodically, as new documents are added to the collection and can reside outside of the document retrieval system, allowing for it to be plugged into any such system, or for the underlying retrieval system to be changed. It is also in contrast to previous approaches to document categorization for retrieval, such as those based on clustering, e.g. clustering by committee (Lin and Pantel, 2002) or semantic class induction as in (Lin and Pantel, 2001b), LDA-based topic modeling (Blei et al., 2003; Griffiths and Steyvers, 2004; Tuarob et al., 2013) and supervised or active learning approaches (Kholghi et al., 2015a) for concept extraction in information retrieval.

2 Background and Motivation

The problem of vocabulary mismatch in information retrieval where semantic overlap may exist while there is no lexical overlap, can be greatly alleviated by the use of query expansion (QE) techniques; whereby a query is reformulated to improve retrieval performance and obtain additional relevant documents by expanding the original query with additional relevant terms, and re-weighting the terms in the expanded query (Xu and Croft, 2000; Riviere et al., 2014). This can also be done by learning semantic classes or related candidate concepts in the text and subsequently tagging documents or content with these semantic concept tags, that could then serve as a means for either query-document keyword matching, or for query expansion, to facilitate downstream retrieval or question answering tasks (Lin and Pantel, 2002; Xu and Croft, 2000; Lin and Pantel, 2001b; Xu et al., 2014; Bhagat and Ravichandran, 2008; Li et al., 2011; Tuarob et al., 2013; Halpin et al., 2007; Lin and Pantel, 2001a; McAuley and Yang, 2016). This is exactly the approach we adopt in order to achieve query expansion in an automated, fully unsupervised fashion, using a neural language model for local relevance feedback (Xu and Croft, 2000).

A major problem of approaches like LSA (Deerwester et al., 1990) and LDA–based topic modeling (Blei et al., 2003; Griffiths and Steyvers, 2004) is that they only consider word co-occurrences at the level of documents to model term associations, which may not always be reliable. Furthermore, these are parameterized approaches, where the number of topics $K$ is fixed; and the final topics learnt are available as bags of words or n-grams from which topic labels must yet be inferred by an expert. In contrast, word and phrasal embeddings take into account local co-occurrence information of terms in the top ranked documents retrieved in response to a query (corresponding to the relevance feedback step in IR). This leads to a better modeling of query ver-
sus document term dependencies (Ganguly et al., 2015; Xu and Croft, 2000) lending itself to direct unsupervised extraction of meaningful terms related to a document, and eventually to the query.

Automatic query expansion techniques can be further categorized as either global or local. While global techniques rely on analysis of a whole collection to discover word relationships, local techniques emphasize analysis of the top-ranked documents retrieved for a query (Xu and Croft, 2000; Manning et al., 2009). Global methods include: (a) query expansion/reformulation with a thesaurus or ontology, e.g. WordNet, UMLS (b) query expansion via automatic thesaurus generation, and (c) techniques like spelling correction (Manning et al., 2009). Local methods adjust a query relative to the documents that initially appear to match the query, which is the basic idea behind our language modeling approach to semantic tagging. Basic local methods comprise: (a) relevance feedback, (b) pseudo-relevance feedback, (or blind relevance feedback), and (c) (global) indirect relevance feedback (Manning et al., 2009). Pseudo-relevance feedback automates the manual part of relevance feedback, so that the user gets improved retrieval performance without an extended interaction.

Here, we find an initial set of most relevant documents, then assuming that the top $k$ ranked documents are relevant, relevance feedback is done as before under this assumption. Our proposed method tries to exactly mimic the human user behavior via pseudo-relevance feedback to semantically pre-tag documents that can later aid downstream novel retrieval for direct querying or refined querying. Thus, in our work we combine this local feedback approach with our neural language model, Phrase2VecGLM, as the query mechanism. Using a pseudo-document representation of top-$K$ TFIDF terms for the document as a query into the GLM, we make novel use of Phrase2VecGLM, to semantically tag documents with phrases representative of latent concepts within those documents. This makes the collection more readily searchable by use of these tags for query expansion in downstream IR, particularly helpful in our specific use case of medical information retrieval (Luo et al., 2008; Kholghi et al., 2015b; De Vine et al., 2014; Halpin et al., 2007; Li et al., 2011; Zhang et al., 2016). Additionally, our method treats all queries in our dataset as unseen at test time, on which our actual results and gains are reported.

3 Dataset and Task

The TREC Clinical Decision Support (CDS) task track investigates techniques to evaluate biomedical literature retrieval systems for providing answers to generic clinical questions about patient cases (Roberts et al., 2016), with a goal toward making relevant biomedical information more discoverable for clinicians. For the 2016 TREC CDS challenge, actual electronic health records (EHR) of patients, in the form of case reports, typically describing a challenging medical case, as shown in Figure 1 are used. A case report is, for our purposes a complex query having a specific information need. There are 30 queries in the challenge dataset, corresponding to such case reports, divided into 3 topic types, at 3 levels of granularity Note, Description and Summary text.

The target document collection is the Open Access Subset of PubMed Central (PMC), containing 1.25 million articles consisting of title, keywords, abstract and body sections. In our work, we develop our query expansion method as a blackbox system using only a subset of 100K documents of the entire collection for which human judgments are made available by TREC. This allows us to derive “inferred measures” for Normalized Discounted Cumulative Gain (NDCG) and Precision at 10 (P@10) scores for our evaluation (Voorhees, 2014). However, we evaluate our method on the entire collection of 1.25 million PMC articles on a separate search engine setup using an ElasticSearch (Gormley and Tong, 2015) instance, that indexes this entire set of articles on all available fields. Our unsupervised document tagging method as outlined in Section 4 employs only the abstract field of the 100K PMC articles, for developing the Phrase2VecGLM language model–based document ranking subsequently used in query expansion.

4 Methodology

In our work, a concept is defined as a “candidate term” or “noun phrase” scored by a chosen metric e.g. top-K TFIDF, for downstream use in our algorithm (see Section 4.1 & Algorithm 1). They are used in both, training, as building blocks for unsupervised model creation by first learning a phrasal embedding space on the document collection and
Figure 1: Sample query from the TREC 2016 challenge dataset, representing a clinical note with patient history, at Note, Description and Summary granularity levels.

subsequent construction of the GLM (Section 4.2), and at inference, for semantically concept-tagging documents. At the time of evaluation, concepts refer to either query terms representing a query document (Yang et al., 2009), or concept tags for target documents. Thus our concepts, predominantly noun phrases, vary from a single unigram term to consisting of up to three terms as employed by our phrase-embedding based language model (Section 4.1). Word embedding techniques use the information around the local context of each word to derive the embeddings. We therefore hypothesize that using these embeddings within a language model (LM) could help to derive terms or concepts that may be closely associated with a given document (Ganguly et al., 2015). Then further extending the model to use embeddings of candidate noun phrases, we could leverage such shared contexts for query expansion, despite no lexical overlap between the query and a given document. This could potentially help both: 1) the global context analysis for IR leading to better downstream retrieval performance from direct query expansion, and, 2) the local context analysis from top-ranked documents aiding query refinement for complex query reformulation within a relevance feedback loop (Su et al., 2015; Xu and Croft, 2000).

4.1 Pre-processing corpus for Phrasal GLM

We first pre-process the documents in our collection by lower-casing the text, removing most punctuation, like commas, periods, ampersands etc. keeping however, the hyphens, in order to retain hyphenated unigrams, also keeping semicolons and colons for context. We use regular expressions to retain periods that occur within a decimal value replacing these with the string decimal that then gets its own vector representation.

Since we implement both unigram and phrasal embedding–based GLMs, we process the same document collection accordingly, for each. For the unigram model, our tokens are single or hyphenated words in the corpus. For the phrasal model, we do an additional step of iterating through each document in the corpus, extracting the noun phrases in each using the textblob (Loria, 2014) toolkit. This at times gave phrases of up to a length of six, so we only admit ones of size up to three which may include some hyphenated words, to avoid tiny frequency counts. We then plug these extracted phrases back into the documents to obtain a “phrase-based corpus” for training, that has both unigrams and variable-length phrases upto 3-grams, with no tokens repeated for the n-gram processed corpus.

We then pre-compute various document and collection level statistics such as raw counts, term frequencies (phrase frequencies for phrasal cor-
query term $q'$. Thus, by this model we have:

$$\prod_{q' \in q} P(q'|d) = \prod_{q' \in q} \left[ \lambda P(q'|d) + \alpha \sum_{t \in d} \hat{P}_{\text{sim,doc}}(q', t|d) + \beta \sum_{t \in d} \hat{P}_{\text{sim,Col}}(q', t|d) + (1 - \lambda - \alpha - \beta) P(q'|C) \right]$$

(1)

Here $P(q'|d)$ and $P(q'|C)$ are the same as direct term sampling without transformation, from either the document $d$ or collection $C$, by a regular Jeliner-Mercer smoothing-based LM as in Equation (2), when $t = q'$:

$$P(d|q) = \prod_{q' \in q} \lambda \hat{P}(q'|d) + (1 - \lambda) \hat{P}(q'|C) = \prod_{q' \in q} \lambda \frac{tf(q', d)}{|d|} + (1 - \lambda) \frac{cf(q')}{|C|}$$

(2)

However, when $t \neq q'$ we may sample the term $t$ either from document $d$ or collection $C$ where the term $t$ is transformed to $q'$. When $t$ is sampled from $d$, since the probability of selecting a query term $q'$, given the sampled term $t$, is proportional to the similarity of $q'$ with $t$, where $\text{sim}(q', t)$ is the cosine similarity between the vector representations of $q'$ and $t$, and $\sum (d)$ is the sum of the similarity values between all term pairs occurring in document $d$, the document term transformation probability can be estimated as:

$$\hat{P}_{\text{sim,doc}}(q', t|d) = \frac{\text{sim}(q', t)}{\sum (d)} \cdot \frac{tf(t, d)}{|d|}$$

(3)

Similarly when $t$ is sampled from $C$, where for the normalization constant, instead of considering all $(q', t)$ pairs in $C$, we restrict to a small neighbourhood of say 3 terms around the query term $q'$, i.e. $N_{q'}$, to reduce the effect of noisy terms, then the collection term transformation probability can be estimated as:

$$\hat{P}_{\text{sim,Col}}(q', t|d) = \frac{\text{sim}(q', t)}{\sum N_{q'}} \cdot \frac{cf(t)}{|C|}$$

(4)

Equation 1 combines all these term transformation events by denoting the probability of observing a query term $q'$ without transformation (standard LM) as $\lambda$, that of document sampling–based transformation as $\alpha$ and the probability of collection sampling–based transformation as $\beta$. 

4.2 Phrasal Embedding-based GLM

Standard Jelinek–Mercer smoothing–based language models used for query–document matching can lead to poor probability estimation when query terms do not appear in the document due to a key independence assumption in these models, wherein query terms are sampled independently from either the document or the collection (Zhai and Lafferty, 2004). Thus given our goal of alleviating vocabulary mismatch to reformulate complex queries, we find that the word-embedding based generalized language model due to Ganguly et al. (2015), that models term dependencies using vector embeddings of terms, lends itself exactly for this purpose as it relaxes this independence assumption to incorporate term similarities via vector embeddings. This leads to better probability estimations in the event of semantic overlap between query terms and documents while no lexical overlap by proposing a generative process in which a “noisy channel” may transform a term $t$ sampled from a document $d$ or the collection $C$, with probabilities $\alpha$ and $\beta$ respectively, into a
Thus, per Equations (2) and (1), deriving the posterior probabilities \( P(d|q) \) for ranking documents with respect to a query involves maximizing the conditional log likelihood of the query terms in a query \( q \) given the document \( d \), as shown:

\[
P(d|q) = -\sum_{q' \in q} \log(P(q'|d))
\]

We use their original word (uni-gram) embedding–based model as a baseline in our work. Our model, Phrase2VecGLM, further augments the original model using variable length noun-phrases in the vocabulary prior to learning the embedding space for the GLM. While the model by Ganguly et al, is designed as an IR matching function, we extend this model in our work to incorporate embeddings of candidate noun phrases from the collection, and re-purpose the model to be used as a pseudo-relevance feedback function to select new query expansion terms (Xu and Croft, 2000). Thus, working with the hypothesis that concepts in the form of “candidate noun-phrases” provide more support for meaning, we update the vocabulary to include noun-phrases of up to a length of three, extracted from the text. The vocabulary terms now consist of phrases, introducing more contextually meaningful terms into the set used in term similarity calculations (Equation 3). This improves concept matching, giving additional coverage toward final query term expansion via LM–based document ranking.

5 Algorithm

Our algorithm (Algorithm 1) works by intrinsically using the Phrase2VecGLM model (Section 4.2) for query expansion, to discover concepts that are similar in the shared local contexts that they occur in, within documents ranked as top-K relevant to a query document, and using one of two options for specified threshold criteria to tag the document, as described below. Thus our algorithm consists of two main parts: 1) A document scoring and ranking module applying directly the phrasal embeddings–based general language model described in sections 4.2, 5.1 & algorithm 1, and, 2) A concept selection module to tag the query document with, coming from the set of top ranked matching documents to a query document from step 1. There are a couple of different variations implemented for the concept selection scheme: (i) Selecting the top TF-IDF term from each of the top-K matching documents as the set of diverse concepts, representative of the query document, and (ii) Selecting the top-similar concept terms matching each of the representative query document terms, using word2vec/Phrase2Vec similarities on the top-ranked set of documents (Mikolov et al., 2013). The code for the corpus pre-processing, model building and inference (semantically tagging documents) is made available online ¹ and the dataset is available publicly ².

5.1 Implementation Details

In the pseudocode given by Algorithm 1, \( < \text{docStats} > \) represents a set of tuples containing various pre-computed document level frequency and similarity statistics, having elements like \( \text{docTermsFreqsRawCounts}, \text{docTermsTFIDFs}, \text{docTermPairSimilaritySums} \). \( < \text{collStats} > \) represents a similar set for collection level frequency and similarity measures with elements like \( \text{collTermsFreqsRawCountsIDFs}, \text{collTermPairSimilaritySums} \). The procedure also assumes available, the precomputed hashtable \( dqTerms \), holding the top TF-IDF terms for each document \( d \), used for querying into the GLM. We have excluded the implementation details for the methods \( \text{selectConceptsEmbeddingsModel}, \text{selectConceptsTFIDF} \) and also the GLM method (which essentially computes Equations (1) and (5) for the query document to be tagged with concepts.

6 Experimental Setup

We run two different sets of experiments: (1) Direct query expansion of the 30 queries in the TREC dataset, using UMLS concepts (Manual, 2008) for our augmented baselines, and, (2) Feedback loop–based query expansion where we use the concept tags for a subset of the top returned articles for the Summary Text–based queries ran against an ElasticSearch index, as query expansion terms, (here MeSH terms-based QE (Adams and Bedrick, 2014) is an augmented baseline), and evaluate both types of runs against our ElasticSearch (ES) index setup described in Section 6.2.

¹https://github.com/manirupa/Phrase2VecGLM
²http://www.trec-cds.org/2016.html#documents
Algorithm 1: Document Ranking and Concept Selection by Phrase2VecGLM

```
Initialize hashtables rankedListBestMatchedDocs, word2vecConcepts, TFIDFConcepts;  
These hold ranked document matches and selected concept tags for documents \( d \in C \);

1: procedure GENERATEDOCUMENTRANKINGSCONCEPTS(queryDocs, vectorEmbeddingsModel, \(<\ 
\text{docStats} >, < \text{collStats} >, \lambda, \alpha, \beta, \text{query length}, K) \n2: for \( d \in \text{queryDocs} \) do
3: rankedListBestMatchedDocs[\[d]] = Phrase2VecGLM(dqTerms[\[d]], \text{query length}, \lambda, \alpha, \beta, \text{docStats}
4: \(<\ 
\text{collStats} >)
5: word2vecConcepts[\[d]] = \n6: selectConceptsEmbeddingsModel(dqTerms[\[d]], \text{docStats})
7: rankedListBestMatchedDocs[\[d]] = \n8: TFIDFConcepts[\[d]] = \n9: selectConceptsTFIDF(dqTerms[\[d]], \text{docStats})
10: rankedListBestMatchedDocs[\[d]], K) \n11: end for
12: end procedure

For direct query expansion we take all granularity levels of query topics described in Section 3, i.e. Summary, Description and Notes text, and feed these into our GLMs obtaining the top-K ranked documents for each query and drawing our query expansion concept tags from this set according to the algorithm described in Section 5. For our augmented query baselines, we use UMLS terms within the above query texts generated from the UMLS Java Metamap API that is quite effective in finding optimal phrase boundaries (Bodenreider, 2004; Chen et al., 2016).

For the relevance feedback–based query expansion, we take the top 10-15 documents returned by our ES index setup for each of the Summary Text queries and use the concept tags assigned to each of these top returned documents by our unigram and phrasal GLMs as the concept tags for query expansion for the original query. We then re-run these expanded queries through the ES search engine to record the retrieval performance. The MeSH terms used for the augmented baseline for the feedback loop case, are directly available for a majority of the PMC articles from the TREC dataset itself. Section 4.1 outlines the details of how the dataset was processed to generate the vocabulary and various elements of the GLM.

6.1 Human–Judged Query Annotation

Additionally, to evaluate our feedback loop method against a human judgments–based baseline, we use Expert Term annotations for the query topics available from a 2016 submission to TREC CDS, where 3 physicians were invited to participate in a manual query expansion experiment. Each physician was assigned 10 out of the 30 query topics from the 2016 challenge. Based on the clinical note, each physician provided a list of 2 to 4 key-phrases. The key-phrases did not have to be part of the note, but could be derived from the physician’s knowledge after reading the note (Chen et al., 2016). The search keywords for the query topics thus manually provided by these domain experts, were used to retrieve corresponding matching PMC article IDs from the PubMed domain. The expert then spot-checked the top-ranked articles to see if these were mostly relevant. If so, they finalized the keywords assigned. Otherwise, they kept fine-tuning the keywords, until they got a desired set of results, simulating exactly the adaptive decision support (relevance feedback loop) in IR. We also develop an interpolated model with a coefficient \( \gamma \) that interpolates between the unigram and phrasal models, which gets performance comparable to the phrasal model, but does not outperform the other models by itself, hence we do not report those results here. Because the challenge data provides relevance judgments only on a subset of documents (which Phrase2VeGLM is trained on), we report our results using the inferred measures (Voorhees, 2014), for “normalized discounted cumulative gain” (NDCG) and “Precision at 10” (P@10). Although the TREC CDS 2016 query set is categorized into three topic types for Diagnosis, Tests and Treatment, we do not divide our evaluation runs into three corresponding sets, evaluating our method’s perfor-
mance on the entire TREC query data set instead.

6.2 Evaluation on ElasticSearch (BM25)

For the search engine–based evaluation of our proposed method, we replicated an ElasticSearch (ES) instance setup with similar settings used in a 2016 challenge submission (Chen et al., 2016). Among the different algorithms available, BM25 (with parameters k1=3 and b=0.75) was selected as the ranking algorithm in our setup due to slightly better performance observed than others, with a logical OR querying model implemented, and the minimum percentage match criterion in ES, for search queries, set at 15% of the keywords matched for a document. Since our GLM outlined in Section 4.2 uses the abstract field of the article for query expansion, we boosted the abstract field 4 times and the title field 2 times in our ES search index setup.

6.3 Results and Discussion

Table 1 outlines our results obtained with the various experimental runs described in Section 6. The hyper–parameters for our best performing models were empirically determined and set to be at \((\lambda, \alpha, \beta) = (0.2, 0.3, 0.2)\) for the word embedding–based GLM and \((\lambda, \alpha, \beta) = (0.2, 0.4, 0.2)\) for the phrasal embedding–based GLM, similar to those reported by Ganguly et al., (2015). All models were evaluated for statistical significance against the respective baselines using a two-sided Wilcoxon signed rank test, for \(p < 0.01\), indicated by bold face value, if found to be significant.

As seen from the results, our unigram and phrasal GLM–based methods for query expansion appear quite promising for both direct query expansion and feedback loop based decision support. For both methods, our trivial baseline is the BM25 algorithm of ElasticSearch itself, that uses only the Summary text from the clinical note as the query, with no expanded set of terms.

We summarize our key findings as follows: We run two additional baselines for generation of QE terms: (i) a vanilla language model using standard Jelinek-Mercer smoothing, equivalent to Phrase2VecGLM with settings \((\lambda, \alpha, \beta) = (0.5, 0.0, 0.0)\) such that the embedding space is not used to derive term similarities, and (ii) the standard Phrase2vec embedding space model itself (De Vine et al., 2014) prior to deriving the GLM. Both these baselines actually perform worse than the trivial BM25 baseline for QE on the Summary text, in both direct and relevance feedback settings.

For direct query expansion, UMLS concepts found within the Summary, Description and Notes text of the query itself, were used as augmented baselines. Of these, the Notes UMLS–based expansion worked rather poorly (we attribute this to extra noise concepts in the lengthy Notes text). Though Description text–based UMLS terms did worse than our vanilla Summary text baseline, the Description UMLS terms run through the unigram GLM to get expanded terms did significantly better than Description UMLS terms indicating that our method helps improve term expansion. For direct query expansion, the biggest gain against the baseline was observed for the Summary text UMLS terms run through the unigram GLM to get expanded terms, with a P@10 value of 0.2817. The phrasal model did comparably to the unigram model, however did not beat it, for the direct setting of query expansion.

For the feedback loop based query expansion method, we had two separate human judgment–based baselines, one using the MeSH terms available from PMC for the top 15 documents returned in a first round of querying the ES index with Summary text, and the other based on the expert annotations of the 30 query topics as described in Section 6. The MeSH terms baseline got a P@10 of 0.2294, even less than our vanilla Summary Text baseline with no expanded terms, while our Expert Terms baseline beat this baseline significantly. One reason for the lower performance of the MeSH terms model, we believe, is lack of MeSH term coverage for all the documents chosen. Our unigram GLM–based expanded terms from the top–15 documents returned by Summary Text beat the Expert Terms baseline quite significantly with P@10 of 0.2792. This was outperformed by the phrasal GLM–based expanded terms model with P@10 of 0.2872.

Finally our combined model using the unigram + phrasal GLM terms from the top–15 off of the Summary text, beat our vanilla baseline, and was outperformed by our very best combined terms model which generated unigram + phrasal GLM–based terms for the top–15 documents for each query, off of the Summary + Summary UMLS concepts, getting a P@10 of 0.3091. As an example to illustrate, a set of concept tags learned by our unigramGLM model may look like:
<table>
<thead>
<tr>
<th>Query Expansion Method</th>
<th>Metric</th>
<th>Query Text</th>
<th>NDCG **</th>
<th>P@10 **</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Direct setting:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BM25+Standard LM (Jelinek-Mercer sm.) QE Terms (baseline)</td>
<td>Summary</td>
<td>0.0475</td>
<td>0.1172</td>
<td></td>
</tr>
<tr>
<td>BM25+Phrase2Vec (without GLM) QE Terms (baseline)</td>
<td>Summary</td>
<td>0.0932</td>
<td>0.2267</td>
<td></td>
</tr>
<tr>
<td>BM25+DescUMLS QE Terms (augmented baseline)</td>
<td>Summary</td>
<td>0.1010</td>
<td>0.2414</td>
<td></td>
</tr>
<tr>
<td>BM25+None (baseline)</td>
<td>Summary</td>
<td>0.1060</td>
<td>0.2489</td>
<td></td>
</tr>
<tr>
<td>BM25+SumUMLS QE Terms (augmented baseline)</td>
<td>Summary</td>
<td>0.1466</td>
<td>0.2644</td>
<td></td>
</tr>
<tr>
<td>BM25+SumUMLS+unigramGLM QE Terms (model)</td>
<td>Summary</td>
<td>0.1387</td>
<td>0.2817</td>
<td></td>
</tr>
<tr>
<td><strong>Feedback Loop setting:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BM25+Standard LM (Jelinek-Mercer sm.) QE Terms (baseline)</td>
<td>Summary</td>
<td>0.0265</td>
<td>0.0867</td>
<td></td>
</tr>
<tr>
<td>BM25+Phrase2Vec (without GLM) QE Terms (baseline)</td>
<td>Summary</td>
<td>0.0662</td>
<td>0.1318</td>
<td></td>
</tr>
<tr>
<td>BM25+MeSH QE Terms (baseline)</td>
<td>Summary</td>
<td>0.0970</td>
<td>0.2294</td>
<td></td>
</tr>
<tr>
<td>BM25+None (baseline)</td>
<td>Summary</td>
<td>0.1060</td>
<td>0.2489</td>
<td></td>
</tr>
<tr>
<td>BM25+Human Expert QE Terms (augmented baseline)</td>
<td>Summary</td>
<td>0.1029</td>
<td>0.2511</td>
<td></td>
</tr>
<tr>
<td>BM25+unigramGLM QE Terms (model)</td>
<td>Summary</td>
<td>0.1173</td>
<td>0.2792 *</td>
<td></td>
</tr>
<tr>
<td>BM25+Phrase2VecGLM QE Terms (model)</td>
<td>Summary</td>
<td>0.1159</td>
<td>0.2872 *</td>
<td></td>
</tr>
<tr>
<td><strong>Feedback Loop Combined Models</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BM25+unigramGLM Terms+Phrase2VecGLM Terms (baseline)</td>
<td>Summary</td>
<td>0.1057</td>
<td>0.2756</td>
<td></td>
</tr>
<tr>
<td>BM25+SumUMLS+unigramGLM Terms+Phrase2VecGLM QE Terms (model)</td>
<td>Summary</td>
<td>0.1206</td>
<td>0.3091 *</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Results for IR after Query Expansion (QE) by different methods using unigram and phrasal GLM–generated QE terms, in direct and feedback loop settings. Bold face values indicate statistical significance at $p < 0.01$ over the previous result or baseline. Single asterisks indicate our best performing models. Double asterisks indicate inferred measures (Voorhees, 2014). Numbers are from evaluation of ranking results based on document relevance judgments available for all 30 queries in the dataset.

In this work, we demonstrate that our proposed method of semantic tagging for query expansion, via word and phrasal GLM–based document ranking for pseudo-relevance feedback, can prove an effective means to serve complex, specific information needs such as clinical queries in medical information retrieval that require adaptive decision support, performing better in some cases than even human expert–provided query expansion terms. This is especially helpful to solve the problem of lack of keyword coverage for all documents in any collection, e.g. MeSH terms for PMC articles. In future we hope to leverage end-to-end recurrent neural architectures such as LSTMs, possibly with attention mechanisms (Rocktäschel et al., 2015; Bahdanau et al., 2014) to improve our current method of semantic tagging for complex querying in medical IR.

**Acknowledgments**

The authors would like to thank Alan Ritter, whose invaluable feedback helped to significantly improve portions of evaluation and presentation of this work, and our collaborators at Nationwide Children’s Hospital whose valuable time, support and resources made this work possible. We also thank our anonymous reviewers for their feedback.
References


Yu Su, Shengqi Yang, Huan Sun, Mudhakar Srivatsa, Sue Kase, Michelle Vanni, and Xifeng Yan. 2015. Exploiting relevance feedback in knowledge graph search. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM.


