UMass at BioASQ 2014: Figure-inspired Text Retrieval

Jesse Lingeman and Laura Dietz

School of Computer Science, University of Massachusetts, Amherst
{lingeman,dietz}@cs.umass.edu

Abstract. Building on our experience with retrieval of figures, figure summarization with sentences from text, we study the utility of figure-based features and techniques for text retrieval. Figure based approaches are compared to approaches using abstracts instead of figures. We also explore two different relevance models: one built using the Unified Medical Language System (UMLS) and one built using Wikipedia.
We conduct several experiments exploring different feature combinations using a model built with the TREC Genomics track for submission to the 2014 BioASQ competition.

1 Introduction

The BioASQ competition is about answering biomedical questions by extracting information from research publications on Pubmed. BioASQ offers several subtasks to participate in: retrieving Pubmed documents that contain an answer, retrieving snippets from those documents that contain an answer, retrieving relevant concepts or RDF triples, and extracting the answer from all retrieved material.

In a cooperation between the Center for Intelligent Information Retrieval and UMass Amherst and the BioNLP group at UMass Medical school in Worcester, we developed a figure-inspired text retrieval method as a new way of retrieving documents and text passages from biomedical publictions. Our method is based on the insight that for biomedical publications, the figures play a central role up to the point where their caption and references provide abstract-like summaries of the paper. In this work we build on our experience with figure summarization and figure ranking algorithms [5,8,1].

We are test driving our figure-inspired retrieval method in the BioASQ competition, where we focus our participation on document and snippet retrieval. As figures are the center of our attention, our methods rely on the availability of full text, e.g. in PMC format. Therefore we only retrieve documents and snippets contained in Pubmed Central. We notice that the available training data covers Pubmed Central only sparsely. Most queries in the gold standard contain just one publication from Pubmed Central; only 13 queries contained at least 10 documents in Pubmed Central. Since it is infeasible to define a complete gold standard ahead of time, our mission is to identify new material from PMC

Table 1. Examples of relevant snippets in PMC.

5319ac18b166e2b806000030 Is clathrin involved in E-cadherin endocytosis?
plasma membranes we have found here that non-trans-interacting e-cadherin is constitutively endocytosed like integrin ligand-independent endocytosis that the formation of **endocytosed vesicles of e-cadherin is clathrin dependent** and that e-cadherin but not other cams at ajs and tjs including nectins claudins and occludin is selectively sorted into the endocytosed (PMC 15263019)
5319abc9b166e2b80600002d Is Rac1 involved in cancer cell invasion?
cells was clearly demonstrated by rna interference assay **rac1 depletion significantly suppressed the frequency of invasion in both quiescent and igf-i-stimulated mda-mb-231 cells** this indicates the necessity of rac1 for igf-i-induced cell invasion in the cells overexpression of rac1 has been (PMC 21961005)

that answers the questions. To demonstrate the existence of relevant material we show examples of relevant snippets in Table 1 and provide more examples in the result section.

In the absence of suitable training data on full documents, we develop and train our method on data from TREC Genomics track 2006 and 2007. Like Bioasq Task 2b(phase A), the Genomics TREC task focuses on retrieving relevant documents and snippets for biomedical questions. The distinctions lie in the use of the Highwire corpus. After training supervised models on the TREC data, they are applied to questions posed in the BioASQ competition.

Our approach takes an Information Retrieval perspective on the problem. First, query expansion is performed with information from UMLS, Wikipedia, and Figures to enrich the question. Second, a ranking of full documents and snippets is retrieved from a corpus of articles from Pubmed Central. Third, we extract features for each document and snippet that indicate its relevance for the question and re-rank document/snippets with a supervised learning-to-rank approach.

2 Background: Information Retrieval

This section introduces document retrieval models and query expansion techniques.

2.1 Sequential Dependence Model

An early IR method called QUERY LIKELIHOOD employed an independence assumption within query terms to score documents with Dirichlet collection smoothing. For query terms $q_1, q_2, ...q_m$, each document D in the collection is scored by a product of scores under each query term.

$$score_{\text{uni}\,(q_1,q_2,...q_m)}(D) = \log \prod_{i=1}^{m} \frac{\#(q_i, D) + \mu \frac{\#(q_i,\cdot)}{\#(\cdot,\cdot)}}{\#(\cdot, D) + \mu} \tag{1}$$

We use the notation '$\cdot$' to denote sums over all possible entries. In particular $\#(q_i, D)$ refers to the term frequency of q_i in the given document, $\#(q_i, \cdot)$ refers to the term frequency of q_i in the corpus, and $\#(\cdot, D)$ is the document length and $\#(\cdot, \cdot)$ number of terms in the collection. The scalar μ controls the amount of collection smoothing applied, and is a hyperparameter to be estimated. Good values of μ are in the range of $[500, 5000]$.

The query likelihood model is almost always outperformed by the SEQUENTIAL DEPENDENCE MODEL [6], which also includes exact bigrams and windowed skip-bigrams. The unigram model above can be generalized to arbitrary count statistics, such as occurrences of a bigram "$q_i\ q_{i+1}$" in document D to derive $score_{\text{bi}}$. Furthermore, counting co-occurrences of the two terms q_i and q_{i+1} in any order within a window of 8 terms in the document gives rise to the score under the windowed bigram model $score_{\text{wbi}}$, where the marginal counts in the denominator $\#(\cdot, D)$ are approximated by the document length.

The sequential dependence model combines the scores of the document D under the unigram, bigram and window model as a log-linear model.

$$\begin{aligned} score_{\text{SDM}\,(q_1,q_2,\ldots q_m)}(D) &= \lambda_{\text{uni}} score_{\text{uni}}(D) + \lambda_{\text{bi}} score_{\text{bi}}(D) + \lambda_{\text{wbi}} score_{\text{wbi}}(D) \\ &= < \boldsymbol{\lambda}, \phi(D) > \end{aligned} \tag{2}$$

The sequential dependence model requires setting of hyperparameters λ_{uni},λ_{bi}, λ_{wbi}, and μ, where the λs can be estimated with machine learning.

2.2 Query Expansion

Keyword-based retrieval methods such as query likelihood and sequential dependence fail to retrieve documents that refer to the query terms via synonyms. A solution is to expand the original query $q_1, q_2, \ldots q_m$ with additional terms $t_1, t_2, \ldots t_K$—so-called expansion terms. Methods for predicting expansion terms t_i also provide confidence weights w_i.

An expanded SDM query scores documents D by

$$score_Q(D) = score_{\text{SDM}\,(q_1,q_2,\ldots q_m)}(D) + \omega \cdot \sum_i w_i \cdot score_{\text{uni}\,(t_i)}(D) \tag{3}$$

The expanded retrieval model introduces another hyperparameter ω, which can be estimated along with λ using machine learning.

2.3 Pseudo-relevance Feedback

Additional expansion terms can be derived from external synonym resources or estimated with pseudo-relevance feedback. In pseudo relevance feedback the expansion terms are estimated from the document collection [3]. The approach is based on the assumption that the un-expanded retrieval model obtained high precision in the top ranks, but was lacking recall.

The procedure gathers a feedback ranking $D_1, D_2, ..., D_n$ from the documents from the collection which have the highest score under the un-expanded query, e.g. $score_{\text{SDM}}(D)$.

The next step derives distribution over terms from the feedback documents. This involves taking the score of the document D_i to approximate a relative retrieval probability of D_i compared to the rest of the feedback set.

$$p(D_i|q_1, ..., q_m) = \frac{1}{\sum_{j=1}^{n} \exp score_{\text{SDM}}(D_j)} \exp score_{\text{SDM}}(D_i) \tag{4}$$

In addition, for each feedback document, a distribution over terms is derived as a language model.

$$p(t|D_i) \propto \frac{\#(t, D_i)}{\#(\cdot, D_i)} \tag{5}$$

These two parts are aggregated to estimate the term distribution for expansion. We derive the estimator as a mixture of document-specific language models where the document retrieval probabilitie govern the mixing weights.

$$p(t) = \sum_{i=1}^{n} p(t|D_i)p(D_i|q_1, ..., q_m) \tag{6}$$

The K most probable terms t_i under this distribution, together with weights $w = p(t_i)$ are predicted as expansion terms.

2.4 Learning Hyperparameters

We exploit that a SDM retrieval model with query expansion falls into the family of log-linear models which can be efficiently estimated with a learning-to-rank approach [7]. We represent each document by a feature vector with four entries: the document's score under the unigram model, as well as the bigram, window-bigram, and expansion model. We use the document relevance assessments from the training set to estimate a log-linear learning-to-rank model.

In this work we use the coordinate ascent learner from the RankLib[1] package optimizing for the metric mean-average precision (MAP).

The weights of the optimal learning-to-rank model are also the optimal settings λ_{uni},λ_{bi}, λ_{wbi} and ω for the retrieval model. When the SDM model is expanded with multiple expansion models this learning-to-rank approach can be generalized appropriately.

This reduces the hyperparameters that need to be estimated by grid-tuning to the Dirichlet smoothing μ for SDM, and number of feedback document n and number of expansion terms K for each expansion model.

[1] http://people.cs.umass.edu/~vdang/ranklib.html

3 Retrieval Approaches

In this section we detail how retrieval and query expansion approaches are combined to leverage figure information to derive a first pass of bio-medical text retrieval. We discuss reranking techniques in Section 4. We refer to the target document collection as full documents, as we further extract pseudo-documents for figures and abstract.

3.1 Indexes

From the full documents in the collection, we create different retrieval indexes.

The FULL DOCUMENT INDEX contains the documents in Pubmed Central document collection. The task is to retrieve relevant documents from this collection. The collection is converted into JSON format using the convertion tool provided by the BioASQ organizers. We index the all visible text as-is while preserving character offsets and section information. The document preprocessing uses a special tokenizer that preserves the names of chemical compounds, genes and pathways.

We identify all figures in the original Pubmed central format and extract so-called FIGURE DOCUMENTS for each of them. The figure document includes the caption of the figure, the sentences that reference the figure. In separate fields we also include sentences within a window of one and two sentences away from a figure reference. We use the figure documents for query expansion and feature generation.

In order to compare the expressiveness of figure documents to abstracts, we also create an index of abstracts that we swap in as a replacement for figure documents.

3.2 Document Retrieval

The most basic retrieval method uses the given query Q to obtain a ranking of full documents under the sequential dependence model. This ranking can be output directly [UMass-irSDM], or submitted to a feature-based re-ranking method (described in Section 4).

We can improve the ranking by expanding the original query with expansion terms (to obtain query Q') to derive a ranking the full documents. To expand the query with pseudo-relevance feedback, we have different options. We can employ the figure document index [FigDoc Query Expansion] to retrieve a feedback run, compute term distributions according to the relevance model and expand the query Q. This approach is also applied to the index of abstract documents to derive the method [Abstract Query Expansion].

As an external source of synonyms we can also use Wikipedia. For that we create a full text index of a Wikipedia snapshot from January 2012 which contains articles for different entities, where some are targeting the biomedical domain. We cast the original query to our Wikipedia index and apply standard pseudo-relevance feedback [Wiki Query Expansion].

Alternatively, we expand the query using an external synonym dictionary. In this study we use the Unified Medical Language System (UMLS) [4,2]. We look up all query terms q_i and all query bi-grams $q_i q_{i+1}$ in the UMLS dictionary to build a pool of expansion terms. Prioritizing for terms that are returned by more than one lookup, we identify K expansion terms [UMLS Query Expansion].

In all approaches we learn the SDM parameters λ and expansion weight ω using 25% of the TREC Genomics queries as training data. We tune the hyperparameter μ of the sequential dependence model using grid-tuning on another 25% of the TREC queries as validation data. We select the maximal μ and according λ and ω and keep it fixed for the remainder of the experiment.

3.3 Snippet Retrieval

To participate in the snippet retrieval task, the goal is to break down the relevant documents into passages that are likely to contain the answer. In the field of Information Retrieval this problem is known under the name Answer-Passage Retrieval.

The passage retrieval approach applies the document retrieval model to consecutive text segments inside the document, to create a ranking on the sub-document level. We chose a granularity of 50 words, which are shifted through the document in increments of 25 words. For efficiency reasons we only consider documents in the high ranks for passage retrieval.

For each document, we only consider the highest ranking passage (called Max-Passage) in the following.

4 Feature-based Re-ranking Approaches

The ranking of full documents created by methods in Section 3 can be further improved with a supervised re-ranking approach. We use four main classes of features. IR Features (Table 2) are derived from the retrieval score under the unigram, bigram, windowed bigram, and expansion model. The Fiat Document Features (Table 3) are based on similarity measures between the query and a semi-structured representation of the full document. Figure captions are included in the text, but not regarded in any special way. The Fiat Figure Features (Table 4) are designed to capture similarity of the query to figure-related information available in the semi-structured document. The fourth category are Figure Document Features (Table 5) which are derived by retrieving figure documents (or abstracts), generate features for every figure, and aggregating across figures within the same document. A full list of features can be found in the appendix.

The main idea behind the figure and figure document features is to use figures as a way to easily isolate important text. There is a lot of technical content in articles, such as related work sections or details on the experimental setup, that are not necessarily relevant to the question being asked and can skew search results. Figures and figure-related passages, on the other hand, are usually describing

Table 2. IR Features for Reranking

Feature Name	Type	Description
docscore	IR	Overall score of the document
docrank	IR	Overall rank of the document
docexpscore	IR	Exponentiated score of the document
docrecrank	IR	Reciprocal rank of the document
unidocscore	IR	Unigram model score
unidocrecrank	IR	Unigram model rank
unidocexpscore	IR	Unigram model exponentiated score
unidocrecrank	IR	Unigram model reciprocal rank
bidocscore	IR	Bigram model score
bidocrank	IR	Bigram model rank
bidocexpscore	IR	Bigram model exponentiated score
bidocrecrank	IR	Bigram model reciprocal rank
wbidocscore	IR	Windowed bigram model score
wbidocrank	IR	Windowed bigram model rank
wbidocexpscore	IR	Windowed bigram exponentiated score
wbidocrecrank	IR	Windowed bigram reciprocal rank
expdocscore	IR	Expansion model score
expdocrank	IR	Expansion model rank
expdocexpscore	IR	Exponentiated score of expansion model
expdocrecrank	IR	Reciprocal rank of expansion model
maxpsgscore	IR	Maximum passage score in the document
maxpsgrank	IR	Highest rank of passage in document
maxpsgexpscore	IR	Exponentiated maximum passage score
maxpsgrecrank	IR	Reciprocal of highest ranked passage

Table 3. Document Features for Reranking

Feature Name	Type	Description
abs.in_abstract	Passage	Is passage in abstract?
tbl.tfidf	Passage	TF-IDF between passage and table captions
tbl.query_cover	Passage	Query cover (QC) of referenced table captions
tbl.num_refs	Passage	Number of references to tables in passage
cite.tfidf	Passage	TF-IDF between passage and
cite.query_cover	Passage	QC of sentences with references to citations
cite.num_refs	Passage	Number of citations in passage
allrefs.tfidf	Passage	TF-IDF to text with refs to figures, tables, or citations
allrefs.query_cover	Passage	QC of references in passage to figures, tables, or citations
allrefs.num_refs	Passage	Number of references in this passage
title.tfidf	Document	TF-IDF between the query and the title
title.query_cover	Document	QC of document title
abs.tfidf	Document	TF-IDF between the query and abstract
abs.query_cover	Document	QC of the abstract of the document
fulltxt.tfidf	Document	TF-IDF between query and the full text of the document
fulltxt.query_cover	Document	QC of the full text of the document

Table 4. Figure-Specific Features for Reranking

Feature Name	Type	Description
fig.num_refs	Passage	Number of references to figures in passage
fig.query_cover	Passage	QC all figure-related sentences referenced in psg
fig.query_cover_caption	Passage	QC of figure captions referenced in this passage
fig.tfidf	Passage	TF-IDF to figure related sentences referenced in psg
fig.tfidf_caption	Passage	TF-IDF between query and referenced figure captions
fig.psg_caption_overlap	Passage	Overlap between passage and referenced figure caption
fig.in_caption	Passage	Is this passage inside of a figure caption?
fig.refs.query_cover	Document	QC of figure-related sentences
fig.refs.query_cover_window1	Document	QC 1 sentence window around figure-related sentences
fig.refs.query_cover_window2	Document	QC 2 sentence window around figure-related sentences
fig.refs.tfidf	Document	TF-IDF of figure-related sentences
fig.refs.tfidf_window1	Document	TF-IDF 1 sentence window around figure-related sents
fig.refs.tfidf_window2	Document	TF-IDF 2 sentence window around figure-related sents
fig.cap.query_cover	Document	QC of figure captions in document
fig.cap.tfidf	Document	TF-IDF between query and all figure captions in doc
fig.refs.has_figs	Document	Does this document have figures?
fig.refs.num_figs	Document	Number of figures in document

Table 5. Figure Document Features for Reranking

Feature Name	Type	Description
figdoc.avgscore	FigDoc	Average score of figure documents for a given document
figdoc.avgrank	FigDoc	Average rank of figure documents for a given document
figdoc.figcount	FigDoc	Total number of figure document returned
figdoc.figcount1	FigDoc	Number of figure documents returned at rank 1
figdoc.figcount3	FigDoc	Number of figure documents returned at rank 3
figdoc.figcount5	FigDoc	Number of figure documents returned at rank 5
figdoc.figcount10	FigDoc	Number of figure documents returned at rank 10
figdoc.figcount20	FigDoc	Number of figure documents returned at rank 20
figdoc.figcount50	FigDoc	Number of figure documents returned at rank 50
figdoc.figcount100	FigDoc	Number of figure documents returned at rank 100
figdoc.figcount1000	FigDoc	Number of figure documents returned at rank 1000
figdoc.maxscore	FigDoc	Maximum score of returned figure documents
figdoc.minrank	FigDoc	Minimum rank of returned figure documents
figdoc.avgreciprank	FigDoc	Average reciprocal rank of returned figure documents
figdoc.maxreciprank	FigDoc	Maximum reciprocal rank of returned figure documents

an important finding of the article. Here, we use the index of figure documents to extract features capturing the essence of findings. The query is issues against the FigDoc index and we keep track how many and at which rank we retrieve figures for the respective document. We also keep track whether high ranking figures are referenced from the highest scoring passage, and measure the textual similarity between passage and high ranked captions. This allows to separate the false positives from the true positives: an article may be highly ranked because of something discussed in the related work or future work sections, however an article that may be slightly lower ranked but has relevant figure documents may be the more relevant document.

We also use features considering the document as a whole. We generate binary values for quality indicators, e.g., whether a document has figures, citations, and tables. We also generate features about the passages, such as number of figure references, number of citation references, number of table references, and the sum of all references in a passage. Binary features are also calculated for whether or not a passage is in a figure caption or in a document abstract.

Most of the generated features compare the tokens in the query to the tokens of some part of the document. Two measures are used to do this: Query Cover and TF-IDF. Query Cover is a simple proportion of how many of the query tokens appear in a particular part of the document. TF-IDF is similar, but each token is weighted by how frequent it appears in the corpus. If a token does not frequently appear in the corpus, but appears often in a part of the document, it gets a higher score than if it is a common token in the corpus. These measures are evaluated over different segments of the document: we obtain scores by comparing the query to the document abstracts, sentences in the document that reference a figure, a window of sentences around a figure reference, figure captions, and sentences in the document that reference a citation or table.

5 Experimental Evaluation

We train and validate our methods on test sets of the TREC Genomics track from the years 2006 and 2007. Both test sets make use of a collection of 162,259

Table 6. Overview of different methods used in the TREC Genomics evaluation.

	IR SDM	IR RM	Rerank IR	Rerank Doc	Rerank Fig	Rerank FigDoc	Rerank All	All no RM
FigDoc Query Expansion		X	X	X	X	X	X	
IR Full Docs	X	X	X	X	X	X	X	X
IR Figure Docs						X	X	X
Supervised Re-ranking			X	X	X	X	X	X
Features IR Doc / Passage			X	X	X	X	X	X
Features Full Docs (Text Only)				X	X		X	X
Features Figures from Full Docs					X		X	X
Features Figure Document						X	X	X

documents from 59 biomedical journals published by Highwire Press. The documents are made available as raw HTML with several download errors and partial documents. The 2006 collection comprises 27 queries and the 2007 collection include 35 queries.

In the following, we make use of a development set comprising the union of the first half of queries from both 2006 and 2007 test collections for feature development and hyperparameter tuning. We report results on both the development set and the combined test sets from 2006 and 2007.

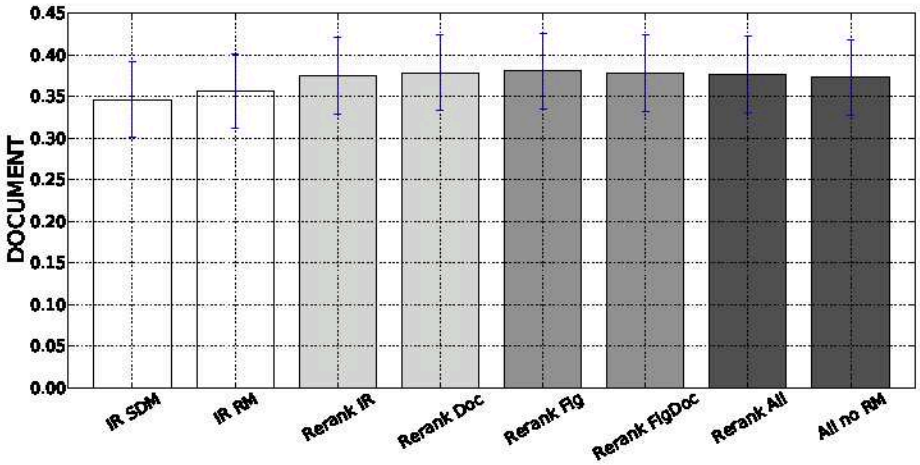

Fig. 1. Cross-validation results on TREC Genomics development set in mean-average precision (MAP).

5.1 Retrieval Hyperparameters

Settings of hyperparameters for retrieval models are determined on the BioASQ training data, which we further subdivide into a 50% training-fold for log-linand a 50% validation-fold. We train the sequential dependence parameters λ_{uni},λ_{bi}, λ_{wbi} and relevance model balance-weight ω in log-linear model fashion with coordinate ascent (using the RankLib package) on the training fold. We tune the Dirichlet smoothing parameter μ on a selection of 100, 1000, 2000, 2500, 3000 on the validation fold.

The parameter settings change with the system. As we aggregate more BioASQ training data from the previous batch submissions (query for task 2b phase b), the parameters also change across batches. A detailed list of which parameter has been used in which batch is given in Table 9.

5.2 Retrieval and Reranking Methods

We study the impact of different components on the overall document retrieval effectiveness, by omitting some components from the pipeline as indicated in Table 7. The most complete method, referred to as "All-Figdoc-UMLS" includes all elements of our pipeline: query expansion on the Figure Document index, retrieval of full documents with the expanded query, generation of various features for re-ranking. The feature sets include scores from the IR system as well as text-only features in addition to figure-related features as extracted from the full documents and Figure Documents.

5.3 Training Supervised Re-ranking on TREC Genomics

As only few BioASQ training queries have more than 10 positive documents in the Pubmed Central collection, we were hesitant to train the supervised re-ranking model on it. We learn the parameter vector for feature-based reranking on the TREC Genomics queries test set, using years 2006 and 2007 on the corpus of Highwire publications. We use 50% of the TREC queries for learning the supervision. As the supervision depends on IR hyperparameters, we apply the tuning heuristic above to 25% of the TREC queries (yielding $\lambda_{\text{uni}} = 0.77$, $\lambda_{\text{bi}} = 0.005$, $\lambda_{\text{wbi}} = 0.037$, $\omega = 0.20$ and $\mu = 2500$).

5.4 Evaluation on TREC Genomics

We study different components of our methods on TREC Genomics holdout set. We evaluate the "Rerank All" method (corresponding to system "All-Figdoc-UMLS") method compared to variants of this approach that omit certain feature classes or steps in the retrieval pipeline. An overview of the evaluated methods is given in Table 6.

The official evaluation metric of the TREC Genomics test set is mean-average precision (MAP) on the document ranking. The results on the development set are presented in Figure 1. We see that the re-ranking approaches gain a decent

boost, whereas the differences between different feature sets are neglegible. With a paired-t-test at significance level $\alpha = 5\%$, we verify that "Rerank All" and "Rerank Doc" yield significant improvements over both IR baselines (despite the overlap in error bars).

5.5 Submission to BioASQ

We restrict all rankings to the top 20 documents, and for each document we provide the best scoring snippet, yielding 20 snippets per system and query. We score snippets with the same retrieval model that we use for document retrieval.

Inspecting all top 50 documents, for each document we create snippet candidates by a sliding window of 50 terms (shifted by 25 terms) and only return the snippet with the highest score under the expanded retrieval model. The snippets are reranked by the retrieval score under the passage model and we only output the top 20 snippets. This means, that some snippets might stem from new documents.

The term windows are converted to section IDs and character offsets. In the batch 1 submission, we did not incorporate whitespaces and XML formatting correctly. This has been corrected for all remaining batches.

Table 7. Overview of different systems submitted to the BioASQ evaluation. 'X' denotes that the component was selected in all batches for this system. Components only selected in some batches are indicated with 'B'.

	UMass-irSDM	Doc-Figdoc-UMLS	All-Figdoc-UMLS	All-Figdoc	All-Abstract-UMLS
FigDoc Query Expansion		X	X	X	
Abtract Query Expansion					X
UMLS Expansion		B1, B2	B1, B2	B1, B2	B1-B5
Wikipedia Expansion		B3, B4, B5	B3, B4, B5	B3, B4, B5	
IR Full Docs	X	X	X	X	X
IR Figure Docs		X	X	X	X
Supervised Re-ranking		X	X	X	X
Features IR Doc / Passage		X	X	X	X
Features Full Docs (Text Only)			X	X	X
Features Figures from Full Docs			X	X	X
Features Figure Document		X	X	X	X

We modified the some components across different submitted batches, to maximize our knowledge gain in the light of the limitation to 5 submission sys-

tems. In particular we varied the query expansion with external sources, from using UMLS to Wikipedia. This change is indicated in Table 7.

Timing. The methods were run on a gridengine cluster each node having a 2.21GHz Intel Xeon CPU with 10GB of RAM (much more than necessary). Averaging the CPU time of 100 queries, we observe 21 seconds for irSDM, 35 seconds for All-FigDoc-UMLS (with Wikipedia Expansion), 41 seconds All-Abstract-UMLS, 25 seconds for All-FigDoc, 36 seconds for Doc-Figdoc-UMLS.

Results. After observing an abysmal score for all our systems on the official preliminary results, we manually inspected the quality of predicted snippets on rank one and two in 25 queries of batch 5 obtained by the irSDM method. Table 10 displays some of the relevant snippets. We notice that many of the documents are not listed in the gold standard. An exception are the query on archeal genomes where we found a much more descriptive snippet than the one provided in the gold standard, and the query on Gray paleted syndrome, where our passage includes the ground truth passage.

We perform a more elaborate annotation on a subset of nine queries from batch 3 (irSDM). The results, measured in snippet precision at rank 10 (P@10) are presented in Table 8. We see that the precision varies between 10% and 70%, but all queries have a non-zero precision. One of our common mistakes occurs when questions ask about a particular brand of medicine or active ingredient. We notice that in such cases, a large percentage of retrieved snippets are about the disease in general, but do not mention the brand or ingredient. In the future, we intend to modify our approach by identifying such required words with an NLP tagger such as conditional random fields and discard snippets that do not contain the required word.

Table 8. P@10 of snippets returned by irSDM on nine selected queries.

Query	P@10
52b2efcb4003448f55000005	0.1
52b2e97df828ad283c000012	0.2
52b2ed144003448f55000004	0.3
52b2ec944003448f55000002	0.6
52b06a68f828ad283c000005	0.7
52b2e409f828ad283c00000e	0.4
52b2ecd34003448f55000003	0.1
52b2e1d8f828ad283c00000c	0.2
52b2f09f4003448f55000008	0.2
average	0.3

6 Conclusion

For the UMass BioASQ submission we designed a figure-aware IR system which includes search-indexes of full document as well as figure captions and references. We use figures both as a resource for query expansion and test external source such as Wikipedia and UMLS as well. The retrieval approach is complemented by a supervised learning-to-rank method the includes features from IR, the document, figure features, and features from retrieving figure documents.

We evaluate against a very strong text-only baseline, which is outperformed on our development test set from the TREC Genomics track. We anticipate that including features from the figure-documents in both the retrieval methods and in reranking will improve the ranking of both document and snippets.

Acknowledgements

This work was supported in part by the Center for Intelligent Information Retrieval, in part by Umass Medical School subaward RFS2014051 under National Institutes of Health grant 5R01GM095476-04. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

1. Agarwal, S., Yu, H.: Figsum: automatically generating structured text summaries for figures in biomedical literature. In: AMIA Annual Symposium Proceedings. vol. 2009, p. 6. American Medical Informatics Association (2009)
2. Bodenreider, O.: The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Research 32(Database issue), D267–D270 (Jan 2004)
3. Lavrenko, V., Croft, W.B.: Relevance based language models. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 120–127. SIGIR '01, ACM, New York, NY, USA (2001), http://doi.acm.org/10.1145/383952.383972
4. Lindberg, D.A., Humphreys, B.L., McCray, A.T.: The Unified Medical Language System. Methods of Information in Medicine 32(4), 281–291 (Aug 1993)
5. Liu, F., Yu, H.: Learning to Rank Figures within a Biomedical Article. PLOS ONE 9(3) (MAR 13 2014)
6. Metzler, D., Croft, W.B.: A markov random field model for term dependencies. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 472–479. SIGIR '05, ACM, New York, NY, USA (2005), http://dx.doi.org/10.1145/1076034.1076115
7. Metzler, D., Croft, W.B.: Linear feature-based models for information retrieval. Inf. Retr. 10(3), 257–274 (Jun 2007), http://dx.doi.org/10.1007/s10791-006-9019-z
8. Yu, H., Liu, F., Ramesh, B.P.: Automatic figure ranking and user interfacing for intelligent figure search. PLoS One 5(10), e12983 (2010)

Table 9. Retrieval parameters used by systems in different batches. Systems that only differ in the re-ranking share the same parameter settings.

	Dirichlet μ	SDM Parameters λ_{uni}, λ_{bi}, λ_{wbi}	RM Weight ω
UMass-irSDM			
Batch 1	3000	0.58, 0.11, 0.11	0.19
Batch 2	2500	0.768, 0.004, 0.036	0.26
Batch 3	2500	0.768, 0.004, 0.036	0.26
Batch 4	2500	0.768, 0.004, 0.036	0.26
Batch 5	3000	0.72, 0.12, 0.16	0.005
Doc-Figdoc-UMLS			
Batch 1	3000	0.58, 0.11, 0.11	0.19
Batch 2	3000	0.58, 0.11, 0.11	0.19
Batch 3	2500	0.768, 0.004, 0.036	0.26
Batch 4	2500	0.768, 0.004, 0.036	0.26
Batch 5	2500	0.768, 0.004, 0.036	0.26
All-Figdoc-UMLS			
Batch 1	3000	0.58, 0.11, 0.11	0.19
Batch 2	3000	0.58, 0.11, 0.11	0.19
Batch 3	2500	0.768, 0.004, 0.036	0.26
Batch 4	2500	0.768, 0.004, 0.036	0.26
Batch 5	2500	0.768, 0.004, 0.036	0.26
All-Figdoc			
Batch 1	2500	0.768, 0.004, 0.036	0.26
Batch 2	2500	0.768, 0.004, 0.036	0.26
Batch 3	2500	0.768, 0.004, 0.036	0.26
Batch 4	2500	0.768, 0.004, 0.036	0.26
Batch 5	2500	0.768, 0.004, 0.036	0.26
All-Abstract-UMLS			
Batch 1	NA	NA	NA
Batch 2	3000	0.56, -0.04, 0.04	0.36
Batch 3	3000	0.72, 0.12, 0.16	0.005
Batch 4	3000	0.56, -0.04, 0.04	0.36
Batch 5	3000	0.72, 0.12, 0.16	0.005

Table 10. Examples of relevant snippets in PMC found within the top 2.

5319abffb166e2b80600002f Which growth factors are known to be involved in the induction of EMT?
in emt induction additionally non-smad signaling pathways activated by tgf-? and cross-talk with other signaling pathways including **fibroblast growth factor fgf** and tumor necrosis factor-? tnf-? signaling **play important roles in emt promotion induction of emt in tumor stromal cells** by (PMC 22111550, rank 1)
5319ac18b166e2b806000030 Is clathrin involved in E-cadherin endocytosis?
plasma membranes we have found here that non-trans-interacting e-cadherin is constitutively endocytosed like integrin ligand-independent endocytosis that the formation of **endocytosed vesicles of e-cadherin is clathrin dependent** and that e-cadherin but not other cams at ajs and tjs including nectins claudins and occludin is selectively sorted into the endocytosed (PMC 15263019, rank 1)
5319abc9b166e2b80600002d Is Rac1 involved in cancer cell invasion?
cells was clearly demonstrated by rna interference assay **rac1 depletion significantly suppressed the frequency of invasion in both quiescent and igf-i-stimulated mda-mb-231 cells** this indicates the necessity of rac1 for igf-i-induced cell invasion in the cells overexpression of rac1 has been (PMC 21961005, rank 1)
5311bcc2e3eabad021000005 Describe a diet that reduces the chance of kidney stones.
stone promoters and inhibitors **reducing deposition and excretion of small particles of caox from the kidney maintaining the antioxidant environment and reducing the chance of them being retained in the urinary tract number of herbal extracts** and their isolated constituents have also shown (PMC 23112535, rank 1)
for age study on the relationship of an **animal-rich diet with kidney stone formation** has shown that as the **fixed acid content of the diet increases urinary calcium excretion also increases the inability to compensate for animal protein-induced calciuric response may be risk factor** for the (PMC 21369385, rank 2)
530cf4fe960c95ad0c000003 Could Catecholaminergic Polymorphic Ventricular Tachycardia (CPVT) cause sudden cardiac death?
case of catecholaminergic polymorphic ventricular tachycardia introduction in reid et al.1 discovered catecholaminergic polymorphic ventricular tachycardia cpvt **cpvt is known to cause syncope or sudden cardiac death** and the three distinguishing features of cpvt has subsequently been described (PMC 19568611, rank 1)
52fe58f82059c6d71c00007a Do archaeal genomes contain one or multiple origins of replication?
genomes in the genus bacillus such positive correlation cannot be explained by the pure c?u/t mutation bias archaeal genomes **multiple replication origins are typically assumed for archaeal genome replication** multiple origins of replication implies multiple changes in polarity in nucleotide (PMC 22942672, rank 1)
52e204a998d0239505000012 Which is the definition of pyknons in DNA?
processed the sequences of the human and mouse genomes using the previously outlined pyknon discovery methodology see methods section as well as ref and generated the corresponding pyknon sets **by definition each pyknon is recurrent motif whose sequence has minimum length minimum number of intact** (PMC 18450818, rank 1)
52d8494698d0239505000007 Which genes have been found mutated in Gray platelet syndrome patients?
nbeal2 is mutated in gray platelet syndrome and is required for biogenesis of platelet alpha-granules platelets are organelle-rich cells that transport granule-bound compounds to tissues throughout the body platelet ?-granules the most abundant platelet organelles store large proteins that when released promote platelet adhesiveness haemostasis and wound (PMC 21765412, rank 1)
52ce531f03868f1b06000031 Are retroviruses used for gene therapy?
frequently employed forms of gene delivery in somatic and germline gene therapies retroviruses in contrast to adenoviral and lentiviral vectors can transfect dividing cells because they can pass through the nuclear pores of mitotic cells this character of retroviruses make them proper candidates (PMC 23210086, rank 2)