Biomedical question-focused multi-document summarization: ILSP and AUEB at BioASQ3

Prodromos Malakasiotis[1,2], Emmanouil Archontakis[1], Ion Androutsopoulos[1,2], Dimitrios Galanis[2], and Harris Papageorgiou[2]

[1] Dept. of Informatics, Athens University of Economics and Business, Greece
rulller@aueb.gr, man.arcon@gmail.com, ion@aueb.gr
http://nlp.cs.aueb.gr/

[2] Institute for Language and Speech Processing, Research Center 'Athena', Greece
malakasiotis@ilsp.gr, galanisd@ilsp.gr, xaris@ilsp.gr
http://www.ilsp.gr/

Abstract. Question answering systems aim to find answers to natural language questions by searching in document collections (e.g., repositories of scientific articles or the entire Web) and/or structured data (e.g., databases, ontologies). Strictly speaking, the answer to a question might sometimes be simply 'yes' or 'no', a named entity, or a set of named entities. In practice, however, a more elaborate answer is often also needed, ideally a summary of the most important information from relevant documents and structured data. In this paper, we focus on generating summaries from documents that are known to be relevant to particular questions. We describe the joint participation of AUEB and ILSP in the corresponding subtask of the BIOASQ3 competition, where participants produce multi-document summaries of given biomedical articles that are relevant to English questions prepared by biomedical experts.

Keywords: biomedical question answering, text summarization

1 Introduction

Biomedical experts are extremely short of time. They also need to keep up with scientific developments happening at a pace that is probably faster than in any other science. The online biomedical bibliographic database PUBMED currently comprises approximately 21 million references and was growing at a rate often exceeding 20,000 articles per week in 2011.[3] Figure 1 shows the number of biomedical articles indexed by PUBMED per year since 1964. Rich sources of structured biomedical information, like the GENE Ontology, UMLS, or DISEASESOME are also available.[4] Obtaining sufficient and concise answers from this wealth of information is a challenging task for traditional search engines, which instead of answers return lists of (possibly) relevant documents that the experts

[3] Consult http://www.ncbi.nlm.nih.gov/pubmed/.

[4] See http://www.geneontology.org/, http://www.nlm.nih.gov/research/umls/, http://diseasome.eu/.

themselves have to study. Consequently, there is growing interest for biomedical question answering (QA) systems [3, 4], which aim to produce more concise answers. To foster research in biomedical QA, the BIOASQ project constructs benchmark datasets, evaluation services, and organizes international biomedical QA competitions since 2012 [20].[5]

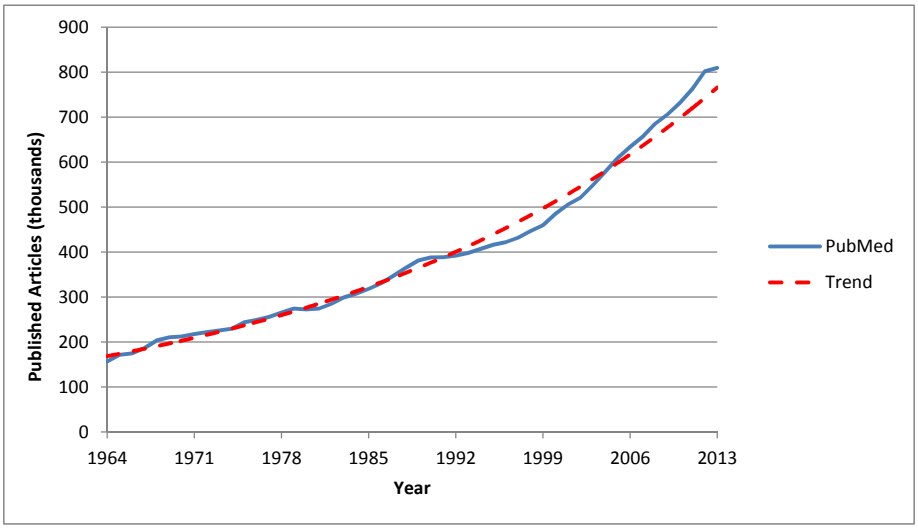

Fig. 1. Number of new PUBMED articles (blue line) indexed over the period 1964-2013 per year, and the respective logarithmic trend (red dashed line).

Given a question expressed in natural language, QA systems aim to provide answers by searching in document collections (e.g., repositories of scientific articles or the entire Web) and/or structured data (e.g., databases, ontologies). Strictly speaking, the answer to a question might sometimes be simply a 'yes' or 'no' (e.g., in biomedical questions like "Do CpG islands co-localize with transcription start sites?"), a named entity (e.g., in "What is the methyl donor of DNA (cytosine-5)-methyltransferases?"), or a set of named entities (e.g., in "Which species may be used for the biotechnological production of itaconic acid?"). Following the terminology of BIOASQ, we call short answers of this kind 'exact' answers. In practice, however, a more elaborate answer is often needed, ideally a paragraph summarizing the most important information from relevant documents and structured data; BIOASQ calls answers of this kind 'ideal' answers. In this paper, we focus on generating 'ideal' answers (summaries) from documents that are known to be relevant to particular questions. We describe our participation in the corresponding subtask of the BIOASQ3 competition (Task 3b, Phase B, generation of 'ideal' answers), where the participants produce summaries of

[5] See also `http://www.bioasq.org/`.

biomedical articles that are relevant to English questions prepared by biomedical experts. In this particular subtask, the input is a question along with the PubMed articles that a biomedical expert identified as relevant to the question; in effect, a perfect search engine is assumed (see Fig. 2). More precisely, in bioasq3 only the abstracts of the articles were available; hence, we summarize sets of abstracts (one set per question). We also note that the abstracts contain annotations showing the snippets (one or more consecutive sentences each) that the biomedical experts considered most relevant to the corresponding questions. We do not use the snippet annotations of the experts, since our system includes its own mechanisms to assess the importance of each sentence. Hence, our system may be at a disadvantage compared to systems that use the snippet annotations of the experts. Nevertheless, experimental results we present indicate that it still performs better than its competitors.

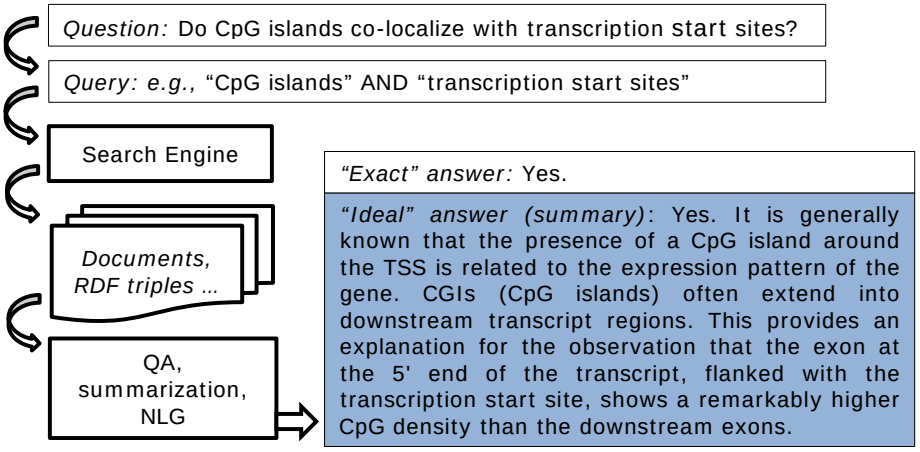

Fig. 2. Using QA, multi-document summarization, and concept-to-text generation to produce 'exact' and 'ideal' answers to English biomedical questions. The blue box indicates the focus of our participation in bioasq3. We did not consider rdf triples.

We also note that when relevant structured information is also available (e.g., rdf triples), concept to text natural language generation (nlg) [1] can also be used to produce 'ideal' answers or texts to be given as additional input documents to the summarizer. We did not consider nlg, however, since in bioasq3 the questions were not accompanied by manually selected (by the biomedical experts) relevant structured information, unlike bioasq1 and bioasq2, and we do not yet have mechanisms to select structured information automatically.

Section 2 below describes the different versions of the multi-document summarizer that we used. Section 3 reports our experimental results. Section 4 concludes and provides directions for future work.

2 Our question-focused multi-document summarizer

We now discuss how the 'ideal' answers (summaries) of our system are produced. Recall that for each question, a set of documents (article abstracts) known to be relevant to the question is given. Our system is an extractive summarizer, i.e., it includes in each summary sentences of the input documents, without rephrasing them. The summarizer attempts to select the most relevant (to the question) sentences, also trying to avoid including in the summary redundant sentences, i.e., pairs of sentences that convey the same information. BIOASQ restricts the maximum size of each 'ideal' answer to 200 words; including redundant sentences wastes space and is also penalized when experts manually assess the responses of the systems [20]. The summarizer does not attempt to repair (e.g., replace pronouns by their referents), order, or aggregate the selected sentences [6]; we leave these important issues for future work.

2.1 Baseline 1 and Baseline 2

As a starting point, we used the extractive summarizer of Galanis et al. [7, 8]. Two versions of the summarizer, known as Baseline 1 and Baseline 2, have been used as baselines for 'ideal' answers in all three years of the BIOASQ competition.[6] Both versions employ a Support Vector Regression (SVR) model [5] to assign a relevance score $rel(s_i)$ to each sentence s_i of the relevant documents of a question q.[7] An SVR learns a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ in order to predict a real value $y_i \in \mathbb{R}$ given a feature vector $\vec{x}_i \in \mathbb{R}^n$ that represents an instance. In our case, $\vec{x}_i$ is a feature vector representing a sentence s_i of the relevant documents of a question q, and y_i is the relevance score of s_i. Consult Galanis et al. [7, 8] for a discussion of the features that were used in the SVR of Baseline 1 and Baseline 2. During training, for each q we compute the ROUGE-2 and ROUGE-SU4 scores [13] between each s_i and the gold (provided by an expert) 'ideal' answer of q, and we take y_i to be the average of the ROUGE-2 and ROUGE-SU4 scores. The motivation for using these scores is that they are the two most commonly used measures for automatic evaluation of machine-generated summaries against gold ones. Roughly speaking, both measures compute the word bigram recall of the summary (or sentence) being evaluated against, possibly multiple, gold summaries. However, ROUGE-SU4 also considers skip bigrams (pairs of words with other ignored intervening words) with a maximum distance of 4 words between the words of each skip bigram. Both measures have been found to correlate well with human judgements in extractive summarization [13] and, hence, training a component (e.g., an SVR) to predict the ROUGE score of each sentence can be particularly useful. Intuitively, a sentence with a high ROUGE score has a high overlap with the gold summaries; and since the gold summaries

[6] Baseline 1 and Baseline 2 are the ILP2 and GREEDY-RED methods, respectively, of Galanis et al. [8]. Baseline 2 had also participated in TAC 2008 [9].

[7] We use the SVR implementation of LIBSVM (see `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`) with an RBF kernel and LIBSVM's parameter tuning facilities.

contain the sentences that human authors considered most important, a sentence with a high ROUGE score is most likely also important.

Baseline 1 uses Integer Linear Programming (ILP) to jointly maximize the relevance and diversity (non-redundancy) of the selected sentences s_i, respecting at the same time the maximum allowed summary length. The ILP model maximizes the following objective function:[8]

$$\max_{b,x} \lambda \sum_{i=1}^{n} \alpha_i \frac{l_i}{l_{max}} x_i + (1-\lambda) \sum_{i=1}^{|B|} \frac{b_i}{n} \tag{1}$$

subject to:

$$\sum_{i=1}^{n} l_i x_i \leq l_{max} \tag{2}$$

$$\sum_{g_j \in B_i} b_j \geq |B_i|\, x_i, \text{for } i = 1, \ldots, n \tag{3}$$

$$\sum_{s_i \in S_j} x_i \geq b_j, \text{for } j = 1, \ldots, |B| \tag{4}$$

where α_i is the relevance score $rel(s_i)$ of sentence s_i normalized in $[0, 1]$; l_i is the word length of s_i; l_{max} is the maximum allowed summary length in words; n is the number of input sentences (sentences in the given relevant documents); B is the set of all the word bigrams in the input sentences; x_i and b_i show which sentences s_i and word bigrams, respectively, are present in the summary; B_i is the set of word bigrams that occur in sentence s_i; g_j ranges over the word bigrams in B_i; and S_j is the set of sentences that contain bigram g_j. Constraint (2) ensures that the maximum allowed summary length is not exceeded. Constraint (3) ensures that if an input sentence is included in the summary, then all of its word bigrams are also included. Constraint (4) ensures that if a word bigram is included in the summary, than at least a sentence that contains it is also included. The first sum of Eq. 1 maximizes the total relevance of the selected sentences. The second sum maximizes the number of distinct bigrams in the summary, in effect minimizing the redundancy of the included sentences. Finally, $\lambda \in [0, 1]$ controls how much the model tries to maximize the total relevance of the selected sentences at the expense of non-redundancy and vice versa. Consult Galanis et al. [7, 8] for a more detailed explanation of the ILP model.

Baseline 2 first uses the trained SVR to rank the sentences s_i of the relevant documents of q by decreasing relevance $rel(s_i)$. It then greedily examines each s_i from highest to lowest $rel(s_i)$. If the cosine similarity between s_i and any of the sentences that have already been added to the summary exceeds a threshold t, then s_i is discarded; the cosine similarity is computed by representing each sentence as a bag of words (using Boolean features), and t is tuned on development

[8] We use the implementation of the Branch and Cut algorithm of the GNU Linear Programming Kit (GLPK); consult `http://sourceforge.net/projects/winglpk/`.

data. Otherwise, if s_i fits in the remaining available summary space, it is added to the summary; if it does not fit, the summary construction process stops.

Baselines 1 and 2 were trained on news articles, as discussed by Galanis et al. [7, 8], and were used in BIOASQ without retraining and without modifying the features of their SVR. However, there are many differences between news and biomedical articles, and many of the features that were used in the SVR of Baselines 1 and 2 are irrelevant to biomedical articles. For example, Baselines 1 and 2 use a feature that counts the names of organizations, persons, etc. in sentence s_i, as identified by a named entity recognizer that does not support biomedical entity types (e.g., names of genes, diseases). They also use a feature that considers the order of s_i in the document it was extracted from, based on the intuition that news articles usually list the most important information first, a convention that does not always hold in biomedical abstracts. Hence, we also experimented with modified versions of Baselines 1 and 2, discussed below, which were trained on BIOASQ datasets and used different feature sets.

2.2 The ILP-SUM-0 and ILP-SUM-1 summarizers

The first new version of our summarizer, called ILP-SUM-0, is the same as Baseline 1 (the baseline that uses ILP, with the same features in its SVR), but was trained on BIOASQ data, as discussed in Section 3 below.

Another version, ILP-SUM-1, is the same as ILP-SUM-0, it was also trained on BIOASQ data, but uses a different feature set in its SVR, still close to the features of Baselines 1 and 2 [7, 8], but modified for biomedical questions and articles. The features of ILP-SUM-1 are the following. All the features of all the versions of the summarizer, including Baselines 1 and 2, are normalized in $[0, 1]$.

(1.1) **Word overlap:** The number of common words between the question q and each sentence s_i of the relevant documents of q, after removing stop words and duplicate words from q and s_i.

(1.2) **Stemmed word overlap:** The same as Feature (1.1), but the words of q and s_i are stemmed, after removing stop words.

(1.3) **Levenshtein distance:** The Levenshtein distance [11] between q and s_i, taking insertions, deletions, and replacements to operate on entire words.

(1.4) **Stemmed Levenshtein distance:** The same as Feature (1.3), but the words of q and s_i are stemmed, before computing the Levensthein distance.

(1.5) **Content word frequency:** The average frequency $CF(s_i)$ of the content words of sentence s_i in the relevant documents of q, as defined by Schilder and Ravikumar [18]:

$$CF(s_i) = \frac{\sum_{j=1}^{c(s_i)} p_c(w_j)}{c(s_i)}$$

where $c(s_i)$ is the number of content words in sentence s_i, $p_c(w) = \frac{m}{M}$, m is the number of occurrences of content word w_j in the relevant documents of q, and M is the total number of content word occurences in the relevant documents of q.

(1.6) **Stemmed content word frequency:** The same as Feature (1.5), but the content words of the relevant documents of q (and their sentences s_i) are stemmed before computing $CF(s_i)$.

(1.7) **Document frequency:** The average document frequency of the content words of sentence s_i in the relevant documents of q, as defined by Schilder and Ravikumar [18]:

$$DF(s_i) = \frac{\sum_{j=1}^{c(s_i)} p_d(w_j)}{c(s_i)}$$

where $p_d(w) = \frac{d}{D}$, d is the number of relevant documents of q that contain the content word w_j, and D is the number of relevant documents of q.

(1.8) **Stemmed document frequency:** The same as Feature (1.7), but the content words of the relevant documents of q (and their sentences s_i) are stemmed before computing $DF(s_i)$.

2.3 The ILP-SUM-2 and GR-SUM-2 summarizers

In recent years, continuous space vector representations of words, also known as word embeddings, have been found to capture several morphosyntactic and semantic properties of words [12, 14–17]. BIOASQ employed the popular WORD2VEC tool [14–16] to construct embeddings for a vocabulary of 1,701,632 words occurring in biomedical texts, using a corpus of 10,876,004 English abstracts of biomedical articles from PUBMED.[9] The ILP-SUM-2 and GR-SUM-2 versions of our summarizer use the following features in their SVR, which are based on the BIOASQ word embeddings, in addition to Features (1.1)–(1.8) of ILP-SUM-1. ILP-SUM-2 also uses the ILP model (like Baseline 1, ILP-SUM-0, ILP-SUM-1), whereas GR-SUM-2 uses the greedy approach of Baseline 2 instead (see Section 2.1).

(2.1) **Euclidean similarity of centroids:** This is computed as:

$$ES(q, s_i) = \frac{1}{1 + ED(\vec{q}, \vec{s_i})} \tag{5}$$

where $\vec{q}$, $\vec{s_i}$ are the centroid vectors of q and s_i, respectively, defined below, and $ED(\vec{q}, \vec{s_i})$ is the Euclidean distance between $\vec{q}$ and $\vec{s_i}$. The centroid $\vec{t}$ of a text t (question or sentence) is computed as:

$$\vec{t} = \frac{1}{|t|} \sum_{i=1}^{|t|} \vec{w}_i = \frac{\sum_{j=1}^{|V|} \vec{w}_j \cdot \mathrm{TF}(w_j, t)}{\sum_{j=1}^{|V|} \mathrm{TF}(w_j, t)} \tag{6}$$

where $|t|$ is the number of words (tokens) in t, and $\vec{w}_i$ is the embedding (vector) of the i-th word (token) of t, $|V|$ is the number of (distinct) words

[9] See `https://code.google.com/p/word2vec/` and `http://bioasq.lip6.fr/tools/BioASQword2vec/` for further details.

in the vocabulary, and $\mathrm{TF}(w_j, t)$ is the term frequency (number of occurrences) of the j-th vocabulary word in the text t.[10]

(2.2) **Euclidean similarity of IDF-weighted centroids:** The same as Feature (2.1), except that the centroid of a text t (question or sentence) now also takes into account the inverse document frequencies of the words in t:

$$\vec{t} = \frac{\sum_{j=1}^{|V|} \vec{w}_j \cdot \mathrm{TF}(w_j, t) \cdot \mathrm{IDF}(w_j)}{\sum_{j=1}^{|V|} \mathrm{TF}(w_j, t) \cdot \mathrm{IDF}(w_j)} \tag{7}$$

where $\mathrm{IDF}(w_j) = \log \frac{|D|}{|D(w_j)|}$, $|D| = 10,876,004$ is the total number of abstracts in the corpus the word embeddings were obtained from, and $|D(w_j)|$ is the number of those abstracts that contain the word w_j.

(2.3) **Pairwise Euclidean similarities:** To compute this set of features (8 features in total), we create two bags, one with the tokens (word occurrences) of the question q and one with the tokens of the sentence s_i. We then compute the similarity $ES(w, w')$ (as in Eq. 5) for every pair of tokens w, w' of q and s_i, respectively, and we construct the following features:

- the average of the similarities $ES(w, w')$, for all the token pairs w, w' of q and s_i, respectively,
- the median of the similarities $ES(w, w')$,
- the maximum similarity $ES(w, w')$,
- the average of the two largest similarities $ES(w, w')$,
- the average of the three largest similarities $ES(w, w')$,
- the minimum similarity $ES(w, w')$,
- the average of the two smallest similarities $ES(w, w')$,
- the average of the three smallest similarities $ES(w, w')$.

(2.4) **IDF-weighted pairwise Euclidean similarities:** The same set of features (8 features) as Features (2.3), but the Euclidean similarity $ES(w, w')$ of each pair of tokens w, w' is multiplied with $\frac{\mathrm{IDF}(w) \cdot \mathrm{IDF}(w')}{\mathrm{MAXIDF}^2}$ to reward pairs with high IDF scores. The IDF scores are computed as in Feature (2.2), and MAXIDF is the maximum IDF score of the words we have embeddings for.

3 Experimental results

We used the datasets of BIOASQ1 and BIOASQ2 to train and tune the four new versions of our summarizer (ILP-SUM-0, ILP-SUM-1, ILP-SUM-2, GR-SUM-2). We then used the dataset of BIOASQ3 to test the two best new versions of our summarizer (ILP-SUM-2, GR-SUM-2) on unseen data, and to compare them against Baseline 1, Baseline 2, and the other systems that participated in BIOASQ3.

[10] Tokens for which we have no embeddings are ignored when computing the features of this section.

3.1 Experiments on BioASQ1 and BioASQ2 data

The BIOASQ1 and BIOASQ2 datasets consist of 3 and 5 batches, respectively, called Batches 1–3 and Batches 4–8 in this section. Each batch contains approximately 100 questions, along with relevant documents, and 'ideal' answers provided by the biomedical experts.

In a first experiment, we aimed to tune the λ parameter of ILP-SUM-0, ILP-SUM-1, and ILP-SUM-2, which use the ILP model of Section 2.1, and compare the three systems. Figure 3 shows the average ROUGE scores of the three systems on Batches 4–6, for different values of λ, using Batches 1–3 to train them (train their SVRs); Batches 7–8 were reserved for another experiment, discussed below. In more detail, we first computed the ROUGE-2 and ROUGE-SU4 scores on Batch 4, training the systems on Batches 1–3 and 5–6. We then computed the average of the ROUGE-2 and ROUGE-SU4 scores of Batch 4, i.e., ROUGE(Batch4) = $\frac{1}{2}$(ROUGE-2(Batch4)+ROUGE-SU4(Batch4)), for each λ value. We repeated the same process for Batches 5 and 6, obtaining ROUGE(Batch5) and ROUGE(Batch6), for each λ value. Finally, we computed (and show in Fig. 3) the average $\frac{1}{3}$(ROUGE(Batch4) + ROUGE(Batch5) + ROUGE(Batch6)), for each λ valuer. Figure 3 shows that ILP-SUM-2 performs better than ILP-SUM-1, which in turn outperforms ILP-SUM-0. The differences in the ROUGE scores are larger for greater values of λ, because greater λ values place more emphasis on the $rel(s_i)$ scores returned by the SVR, which are affected by the different feature sets of the three systems. For $\lambda > 0.8$, the ROUGE scores decline, because the systems place too much emphasis on avoiding redundant sentences. The best of the three systems, ILP-SUM-2, achieves its best performance for $\lambda = 0.8$.

In a second experiment, we compared ILP-SUM-2, which is the best of our new versions that use the ILP model, against GR-SUM-2, which uses the same features, but the greedy approach instead of the ILP model. We set $\lambda = 0.8$ in ILP-SUM-2, based on Fig. 3. In GR-SUM-2, we set the cosine similarity threshold (Section 2.1) to $t = 0.4$, based on Galanis et al. [7, 8]. Figure 4 shows the average ROUGE-2 and ROUGE-SU4 score of each system on Batches 7 and 8, using an increasingly larger training dataset, consisting of Batches 1–3, 1–4, 1–5, or 1–6. A first observation is that ILP-SUM-2 outperforms GR-SUM-2. Moreover, it seems that both systems would benefit from more training data.

3.2 Experiments on BioASQ3 data

In BIOASQ3, we participated with ILP-SUM-2 (with $\lambda = 0.8$) and GR-SUM-2 (with $t = 0.4$), both trained on all 8 batches of BIOASQ1 and BIOASQ2. Baseline 1 and Baseline 2, which are also versions of our own summarizer, were used again as the official baselines for 'ideal' answers, as in BIOASQ1 and BIOASQ2, i.e., without modifying their features or retraining them for biomedical data. The test dataset of BIOASQ3 contained five new batches, hereafter called BIOASQ3 Batches 1–5; these are different from Batches 1–8 of BIOASQ1 and BIOASQ2.

For each BIOASQ3 batch, Table 1 shows the ROUGE-2, ROUGE-SU4, and average of ROUGE-2 and ROUGE-SU4 scores of the four versions of our summarizer

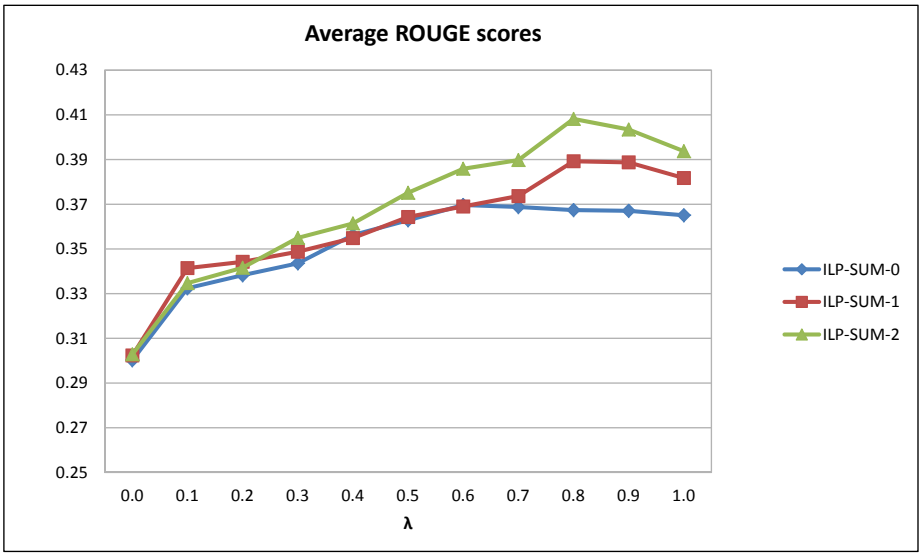

Fig. 3. Average ROUGE-2 and ROUGE-SU4 on Batches 4–6 of BIOASQ1 and BIOASQ2, for different λ values, each time using the five other batches of Batches 1–6 for training.

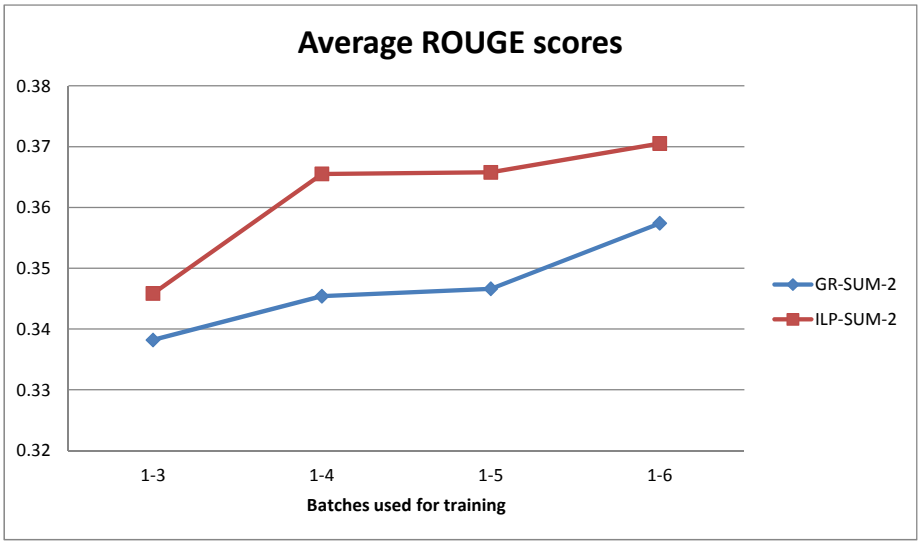

Fig. 4. Average ROUGE-2 and ROUGE-SU4 scores on Batches 7-8 of BIOASQ1 and BIOASQ2, using increasingly more of Batches 1–6 for training.

(ILP-SUM-2, GR-SUM-2, Baseline 1, Baseline 2), ordered by decreasing average ROUGE-2 and ROUGE-SU4. The results of the three other best (in terms of average ROUGE-2 and ROUGE-SU4) participants per batch are also shown, as PART-SYS-1, PART-SYS-2, PART-SYS-3; PART-SYS-1 is not necessarily the same system in all batches, and similarly for PART-SYS-2 and PART-SYS-3.[11] The four versions of our summarizer are the best four systems in all five batches of Table 1.

As in the experiments of Section 3.1, Table 1 shows that ILP-SUM-2 consistently outperforms GR-SUM-2. Similarly, Baseline 2 (which uses the greedy approach) performs better than Baseline 1 (which uses the ILP model) only in the third batch. It is also surprising that ILP-SUM-2 and GR-SUM-2 do not always perform better than Baselines 1 and 2, even though the former systems were tailored for biomedical data by modifying their features and retraining them on the datasets of BIOASQ1 and BIOASQ2. This may be due to the fact that Baseline 1 and Baseline 2 were trained on larger datasets than ILP-SUM-2 and GR-SUM-2 [7, 8]. Hence, training our summarizer on more data, even from another domain (news) may be more important than training it on data from the application domain (biomedical data, in the case of BIOASQ) and modifying its features.

It would be interesting to check if the conclusions of Table 1 continue to hold when the systems are ranked by the manual (provided by biomedical experts) evaluation scores of their 'ideal' summaries, as opposed to using ROUGE scores. At the time this paper was written, the manual evaluation scores of the 'ideal' answers of BIOASQ3 had not been announced.

4 Conclusions and future work

We presented four new versions (ILP-SUM-0, ILP-SUM-1, ILP-SUM-2, GR-SUM-2) of an extractive question-focused multi-document summarizer that we used to construct 'ideal' answers (summaries) in BIOASQ3. The summarizer employs an SVR to assign relevance scores to the sentences of the given relevant abstracts, and an ILP model or an alternative greedy strategy to select the most relevant sentences avoiding redundant ones. The two official BIOASQ baselines for 'ideal' answers, Baseline 1 and Baseline 2, are also versions of the same summarizer; they use the ILP model or the greedy approach, respectively, but they were trained on news articles and their features are not always appropriate for biomedical data. By contrast the four new versions were trained on data from BIOASQ1 and BIOASQ2. ILP-SUM-0, ILP-SUM-1, and ILP-SUM-2 all use the ILP model, but ILP-SUM-0 uses the original features of Baselines 1 and 2, ILP-SUM-1 uses a slightly modified feature set, and ILP-SUM-2 uses a more extensive feature set that includes features based on biomedical word embeddings. GR-SUM-2 uses the same features as ILP-SUM-2, but with the greedy mechanism.

A preliminary set of experiments on BIOASQ1 and BIOASQ2 data indicated that ILP-SUM-2 performs better than ILP-SUM-0 and ILP-SUM-1, showing the importance of modifying the feature set. ILP-SUM-2 was also found to perform

[11] The results of all the systems can be found at `http://participants-area.bioasq.org/results/3b/phaseB/`.

BIOASQ3 Batch 1 (15 systems, 6 teams)			
System	ROUGE-2	ROUGE-SU4	Avg.
ILP-SUM-2	0.4050	0.4213	0.4132
Baseline 1	0.4033	0.4217	0.4125
GR-SUM-2	0.3829	0.4052	0.3941
Baseline 2	0.3604	0.3787	0.3696
PART-SYS-1	0.2940	0.3071	0.3006
PART-SYS-2	0.2934	0.3066	0.3000
PART-SYS-3	0.2929	0.3069	0.2999
BIOASQ3 Batch 2 (16 systems, 6 teams)			
System	ROUGE-2	ROUGE-SU4	Avg.
Baseline 1	0.4657	0.4860	0.4759
Baseline 2	0.4201	0.4493	0.4347
ILP-SUM-2	0.4071	0.4460	0.4266
GR-SUM-2	0.3934	0.4249	0.4092
PART-SYS-1	0.3597	0.3770	0.3684
PART-SYS-2	0.3561	0.3742	0.3652
PART-SYS-3	0.3523	0.3710	0.3617
BIOASQ3 Batch 3 (17 systems, 6 teams)			
System	ROUGE-2	ROUGE-SU4	Avg.
ILP-SUM-2	0.4843	0.5155	0.4999
Baseline 2	0.4586	0.4806	0.4696
GR-SUM-2	0.4482	0.4756	0.4619
Baseline 1	0.4396	0.4661	0.4529
PART-SYS-1	0.3834	0.3950	0.3892
PART-SYS-2	0.3836	0.3941	0.3889
PART-SYS-3	0.3796	0.3906	0.3851
BIOASQ3 Batch 4 (17 systems, 6 teams)			
System	ROUGE-2	ROUGE-SU4	Avg.
Baseline 1	0.4742	0.4947	0.4845
ILP-SUM-2	0.4718	0.4942	0.4830
GR-SUM-2	0.4480	0.4708	0.4594
Baseline 2	0.4345	0.4506	0.4426
PART-SYS-1	0.3864	0.3906	0.3885
PART-SYS-2	0.3606	0.3711	0.3659
PART-SYS-3	0.3627	0.3684	0.3656
BIOASQ3 Batch 5 (17 systems, 6 teams)			
System	ROUGE-2	ROUGE-SU4	Avg.
Baseline 1	0.3947	0.4252	0.4100
ILP-SUM-2	0.3698	0.4039	0.3869
GR-SUM-2	0.3698	0.4039	0.3869
PART-SYS-1	0.3752	0.3945	0.3849
PART-SYS-2	0.3751	0.3910	0.3831
PART-SYS-3	0.3731	0.3930	0.3831
Baseline 2	0.3406	0.3766	0.3586

Table 1. Results of four versions of our summarizer (ILP-SUM-2, GR-SUM-2, Baseline 1, Baseline 2) on the BIOASQ3 batches, along with results of the three other best systems (PART-SYS-1, PART-SYS-2, PART-SYS-3) per batch. Baselines 1 and 2 were not retrained or otherwise modified for biomedical data. ILP-SUM-2 and GR-SUM-2 were trained on the datasets of BIOASQ1 and BIOASQ2. The total numbers of systems and teams that participated in each batch are shown in brackets.

better than GR-SUM-2, which uses the same feature set, showing the benefit of using the ILP model instead of the greedy approach. Our experiments also indicated that ILP-SUM-2 and GR-SUM-2 would probably benefit from more training data. In BIOASQ3, we participated with ILP-SUM-2 and GR-SUM-2, tuned and trained on BIOASQ1 and BIOASQ2 data. Along with Baselines 1 and 2, which are also versions of our own summarizer, ILP-SUM-2 and GR-SUM-2 were the best four systems in terms of ROUGE scores in all five batches of BIOASQ3. Again, ILP-SUM-2 consistently outperformed GR-SUM-2, but surprisingly ILP-SUM-2 and GR-SUM-2 did not always perform better than Baselines 1 and 2. This may be due to the fact that Baselines 1 and 2 were trained on more data, suggesting that the size of the training set may be more important than improving the feature set or using data from the biomedical domain.

Future work could consider repairing, ordering, or aggregating the sentences of the 'ideal' answers, as already noted. The centroid vectors of ILP-SUM-2 and GR-SUM-2 could also be replaced by paragraph vectors [10] or vectors obtained by using recursive neural networks [19]. Another possible improvement could be to use METAMAP [2], a tool that maps biomedical texts to concepts derived from UMLS.[12] We could then compute new features that measure the similarity between a question and a sentence in terms of biomedical concepts.

Acknowledgements

The work of the first author was funded by the Athens University of Economics and Business Research Support Program 2014-2015, "Action 2: Support to Postdoctoral Researchers".

References

1. Androutsopoulos, I., Lampouras, G., Galanis, D.: Generating natural language descriptions from OWL ontologies: the NaturalOWL system. Journal of Artificial Intelligence Research 48, 671–715 (2013)
2. Aronson, A.R.: Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In: Proceedings of the American Medical Informatics Association Symposium. pp. 18–20. Washington DC, USA (2001)
3. Athenikos, S., Han, H.: Biomedical question answering: A survey. Computer Methods and Programs in Biomedicine 99(1), 1–24 (2010)
4. Bauer, M., Berleant, D.: Usability survey of biomedical question answering systems. Human Genomics 6(1)(17) (2012)
5. Drucker, H., Burges, C.J., Kaufman, L., Smola, A., Vapnik, V., et al.: Support vector regression machines. Advances in Neural Information Processing Systems 9, 155–161 (1997)
6. Filippova, K., Strube, M.: Sentence fusion via dependency graph compression. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 177–185. Honolulu, Hawaii (2008)

[12] See `http://metamap.nlm.nih.gov/`.

7. Galanis, D.: Automatic generation of natural language summaries. Ph.D. thesis, Department of Informatics, Athens University of Economics and Business (2012)
8. Galanis, D., Lampouras, G., Androutsopoulos, I.: Extractive multi-document summarization with integer linear programming and support vector regression. In: Proceedings of COLING 2012. pp. 911–926. Mumbai, India (2012)
9. Galanis, D., Malakasiotis, P.: AUEB at tac 2008. In: Proceedings of the Text Analysis Conference. pp. 42–47. Gaithersburg, MD (2008)
10. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on Machine Learning. pp. 1188–1196. Beijing, China (2014)
11. Levenshtein, V.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physice-Doklady 10, 707–710 (1966)
12. Levy, O., Goldberg, Y., Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association for Computational Linguistics 3, 211–225 (2015)
13. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Proceedings of the ACL workshop 'Text Summarization Branches Out'. pp. 74–81. Barcelona, Spain (2004)
14. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of Workshop at International Conference on Learning Representations. Scottsdale, AZ, USA (2013)
15. Mikolov, T., Yih, W., Zweig, G.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the Conference on Neural Information Processing Systems. Lake Tahoe, NV (2013)
16. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies. Atlanta, GA (2013)
17. Pennington, J., Socher, R.and Manning, C.D.: GloVe: Global vectors for word representation. In: Proceedings of the Conference on Empirical Methods on Natural Language Processing. Doha, Qatar (2014)
18. Schilder, F., Kondadadi, R.: Fastsum: Fast and accurate query-based multi-document summarization. In: Proceedings of 46th Annual Meeting of the Association for Computational Linguistics - Human Language Technologies, Short Papers. pp. 205–208. Columbus, Ohio (2008)
19. Socher, R., Huval, B., Manning, C.D., Ng, A.Y.: Semantic compositionality through recursive matrix-vector spaces. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 1201–1211. Jeju Island, Korea (2012)
20. Tsatsaronis, G., Balikas, G., Malakasiotis, P., Partalas, I., Zschunke, M., Alvers, M., Weissenborn, D., Krithara, A., Petridis, S., Polychronopoulos, D., Almirantis, Y., Pavlopoulos, J., Baskiotis, N., Gallinari, P., Artieres, T., Ngonga, A., Heino, N., Gaussier, E., Barrio-Alvers, L., Schroeder, M., Androutsopoulos, I., Paliouras, G.: An overview of the BioASQ large-scale biomedical semantic indexing and question answering competition. BMC Bioinformatics 16(138) (2015)