Word Embeddings and External Resources for Answer Processing in Biomedical Factoid Question Answering

Dimitris Dimitriadis, Grigorios Tsoumakas

School of Informatics, Aristotle University of Thessaloniki, 54124, Greece

Abstract

Biomedical question answering (QA) is a challenging task that has not been yet successfully solved, according to results on international benchmarks, such as BioASQ. Recent progress on deep neural networks has led to promising results in domain independent QA, but the lack of large datasets with biomedical question-answer pairs hinders their successful application to the domain of biomedicine.

We propose a novel machine-learning based answer processing approach that exploits neural networks in an unsupervised way through word embeddings. Our approach first combines biomedical and general purpose tools to identify the candidate answers from a set of passages. Candidates are then represented using a combination of features based on both biomedical external resources and input textual sources, including features based on word embeddings. Candidates are then ranked based on the score given at the output of a binary classification model, trained from candidates extracted from a small number of questions, related passages and correct answer triplets from the BioASQ challenge.

Our experimental results show that the use of word embeddings, combined

Email addresses: `dndimitri@csd.auth.gr` (Dimitris Dimitriadis), `greg@csd.auth.gr` (Grigorios Tsoumakas)

with other features, improves the performance of answer processing in biomedical question answering. In addition, our results show that the use of several annotators improves the identification of answers in passages. Finally, our approach has participated in the last two versions (2017, 2018) of the BioASQ challenge achieving competitive results.

Keywords: biomedical question answering, answer processing, supervised method, word embeddings

2010 MSC: 68T50

1. Introduction

Traditional Information Retrieval (IR) systems, which provide a large amount of documents as potentially relevant results for posed questions, cannot meet the expectations of biomedical researchers and physicians to find instantly the answer they are looking for. The answer is typically buried inside the documents and the questioners must go through the documents to find it. In addition, as the typical input of IR systems is a list of keywords, questioners lose the expressive power of natural language. On the other hand, Question Answering (QA) systems, a sophisticated form of IR systems [1], are capable of providing precise and quick answers to textual questions.

Our work focuses on answering factoid questions which are defined as fact-based, short answer questions [2]. For example, the factoid question "*Treatment of which disease was investigated in the MR CLEAN study?*", can be answered by returning the simple fact "*acute ischemic stroke*". The typical architecture of a factoid QA system comprises three phases [3]. The question processing phase is responsible for analyzing and classifying the question and for converting the question into one or more queries. Next, the document retrieval phase retrieves a set of documents related to the queries and extracts a set of passages containing the answer. Finally, the answer processing (AP) phase extracts a set of candidate answers from passages, ranks them and selects the final answer that is presented to the user.

According to [4], two classes of algorithms are mainly used in AP. In the redundancy-based approach[5], which is mostly applied in web-based QA systems, unigrams, bigrams and trigrams are extracted as candidate answers, then they are ranked based on the answer type and finally, some of them are concatenated in order to create a longer answer. On the other hand, pattern-based extraction methods[6] extract candidate answers based on the answer type together with regular expression patterns. Next, a classifier ranks the list of candidate answers by probability of being correct. We propose a hybrid approach that combines elements from both of these two classes of algorithms. We follow the redundancy-based approach in the answer extraction phase. However, instead of considering every unigram, bigram and trigram as candidate answer, we extract biomedical terms, nouns and numbers. We also follow the supervised learning approach proposed in pattern-extraction methods defining features based on both external resources and the input textual sources, including word embeddings.

External resources can provide semantic information about the elements of the input textual sources. For instance, they can tell us that a question asks about an enzyme and they can identify enzyme terms in the passages. Encoding into one or more features this semantic match between a question and a candidate answer, is expected to improve the performance of answer processing. On the other hand, using information from the given input textual sources, we can extract features that can encode the statistical relationship between the sources. For example, estimating the textual similarity between question and passages, we can find the passage that is closer to the question and is more probable to contain the answer. Finally, word embeddings have been used in most successful deep learning approaches to QA. Enriching the set of features using word embeddings, we expect that the input textual sources will be enriched by additional semantic information learned in an unsupervised manner from large collections of biomedical documents.

Specifically, the main contributions in this paper are:

- A novel representation for the candidate answers of an answer processing module, combining features based on simple statistics, external resources and word embeddings. Our results report that word embeddings, when combined with the rest of the features, improve the overall performance of an AP module.

- A novel empirical study showing that the increasing number of incorrect candidate answers, produced by several general purpose and domain specific annotators, improve the performance of answer processing system despite the extreme class imbalance issue.

- We experimentally evaluate our system in the BioASQ challenge [7] outperforming the BioASQ baseline system and achieving promising results against the other participants.

The rest of the paper is structured as follows. Section 2 reviews related work. Section 3 describes our approach for solving the AP task. Section 4 describes the experimental setup that we used to train and test our approach, while Section 5, presents the results of our approach. Finally, Section 6 concludes this work and points to future research directions.

2. Literature Review

Several different approaches and architectures have been proposed for biomedical AP. Those that are most related to this work correspond to approaches adopted by participants of the BioASQ challenge. Therefore, we discuss here the systems that participated in the BioASQ challenge from 2013, when the challenge started, until 2017. In addition, we discuss recent systems that are based on deep learning in the context of reading comprehension question answering (RCQA), a task that is very similar to AP.

Most of the current systems assume that potential answers to factoid questions are the elements which: (a) can be annotated by general purpose and/or domain specific annotators (e.g. biomedical terms, noun chunks) (b) correspond to specific hand-written patterns (c) do not meet a specific criterion (e.g. all words in related passages). For instance, [8], [9], [10] extract all annotated terms as candidate answers from a set of passages, containing the correct answer, using

MetaMap. [11] use PubTator[1] to identify biomedical terms. They also extract nouns, noun chunks and numbers as candidate answers. Using LingPipe [12] and MetaMap, [13] and [14] extract biomedical terms as candidate answers. They also extract tokens produced by specific patterns and tokens annotated by specific part of speech tags. [15] use PubTator to extract biomedical terms as candidate answers and they also extract numbers. All words in related passages and biomedical terms using MetaMap were extracted as candidate answers in our previous work [16]. [17] extract noun chunks from the related passages. [18] extract answer spans using context/type matching heuristic. [19], using the YodaQA system [20], extract named entities and noun phrases, leveraging information from knowledge bases, or filtering passages.

Different approaches have been proposed in answer ranking phase. [8] use logistic regression and define three classes of hand-written features (Prominence, Type Coercion, Specificity). The produced formula is used to score the candidate answers. [19] learn a classifier using logistic regression and define features emphasizing on type coercion. In our previous works [10], [16], we combine the results of a set of feature functions in order to produce the final score for a candidate answer. [9], [11] rank the candidate answers based on term frequency metrics. [13] use 11 groups of features. In the learning process, they use the questions whose the answers can be produced by their answer extraction approach. They create a dataset where they assign 1 to each candidate answer

[1] https://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/PubTator/

belong to the set of expected correct answers and the candidate answer variants if it is also contained in the gold standard answer set, and 0 otherwise. A similar approach has been adopted by [14] but they also apply a collective answer reranking to boost the candidate answers that are low in the ranking. [15] employ cosine similarity between the candidate answers and the passages. The candidate answer which matches better with the snippets, is extracted as final answer. A similar approach was adopted by [17]. Nevertheless, the latter concatenate the query, produced by the question processing phase, with the candidate answers. Then, they estimate the similarity between the new sequence of tokens with the relevant passages. Furthermore, they estimate the similarity between candidate answers and a set of ideal answers[2] . A different approach was proposed by [18],[21] who extend the FastQA [22] using biomedical word embeddings. The neural model was pre-trained on a large-scale opendomain QA dataset and then the parameters were fine-tuned on the BioASQ training set producing as output start and end pointers to tokens in the relevant passages.

A lot of work that is similar to AP has been conducted in the area of RCQA. AQUAREAS [23] answers questions belonging to one of the five possible types what-, when-, who-, where-questions giving a sentence from a story related with the question. A rule-based Chinese RCQA was proposed by [24] who used heuristic rules to look for lexical and semantic clues in the question and the story. [25] use: (a) metadata to represent questions and answer sentences and

[2] BioASQ defines ideal answers as English paragraph-sized summaries which can be considered as answers

(b) answer patterns derived from question-answer pairs from TREC QA, the Google search engine and the Web. [26] follow a bag-of-words approach combining named entity filtering, pronoun resolution and verb dependency matching. Recently, deep learning approaches have been emerging as state of the art for RCQA. [27] introduced gated self-matching networks for RCQA. They proposed, a four-layer neural network that aims to answer questions from a given passage. Firstly, a bi-directional recurrent network builds representation for passages and questions separately. Next, gated attention-based recurrent networks build a question-aware representation for the passage, matching the question and the passage. In passage self-matching layer, a passage is enriched with additional information aggregating evidence from the whole passage and finally, the output layer predicts the boundary of the answer span. [28] use dynamic coattention networks. Firstly, they build a representation matching relevant parts of the question and the document and then, a dynamic pointing decoder iterates over potential answer spans to predict the boundary of the expected correct answer span. [29] preprocess the passage and the question to incorporate contextual information into the representation of each token. Next, a match-LSTM layer applies textual entailment treating the question as a premise and the passage as a hypothesis. Finally, a pointer network is used to predict the boundary of the answer span. [30] use skip grams to encode the input textual sources and a memory network for matching question and passage catching more vital information. [31] use convolutional units instead of recurrent units achieving comparable results in RCQA. [32] propose a reattention mechanism to avoid the problems of attention redundancy and attention deficiency.

Our approach borrows many ideas from the aforementioned approaches. We use word embeddings to encode input textual sources as most of the current approaches in RCQA do [33]. However, we also concatenate features that leverage biomedical knowledge from external resources, an approach for biomedical AP that was utilized by many researchers in the BioASQ challenge as described earlier in this section. Furthermore, we also extract features from the textual sources, which is another approach used in the BioASQ challenge. The key difference from other approaches is that we combine all the above features into a single representation.

3. Our Approach

Our AP approach comprises three phases (Figure 1). In the answer extraction phase, a list of candidate answers are extracted from the set of passages obtained by the preceding document retrieval phase. Next, the answer representation phase converts candidate answers to multi-dimensional feature vectors. Finally, in the answer ranking phase, a machine learning model scores the candidate answers and outputs the one with the highest score as the expected correct answer.

3.1. Answer Extraction

We follow three steps to extract candidate answers from passages. Firstly, we use regular expressions to detect and remove: (a) web links, (b) all words, mathematical equations and symbols in parentheses, excluding abbreviations,

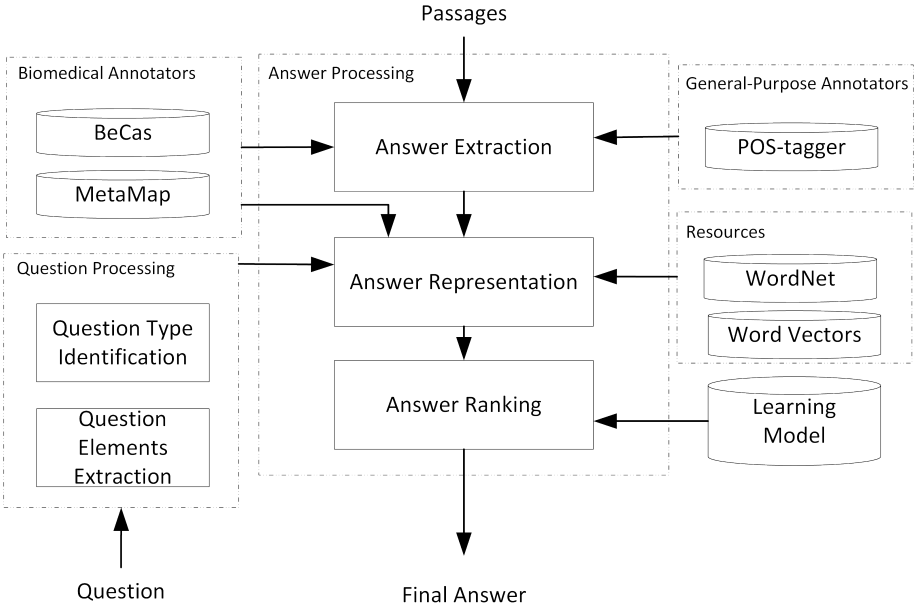


Figure 1: Architecture of our AP approach.

(c) and citations. We assume that a abbreviation is a single token which could contain one or more dashes or numbers. Thus, when we find parenthesis inside the text, we apply the pattern \s*[a-zA-Z0-9-]*\s*$. Next, we split the passages into sentences[3]. Finally, we extract the candidate answers from the sentences.

We assume that candidate answers are nouns, biomedical terms or numbers. We use a general purpose part-of-speech tagger[4] to detect the nouns of each sentence. To extract biomedical terms and numbers, we input each sentence to MetaMap[5] [34] and BeCAS[6] [35]. BeCAS and MetaMap are named entity

[3] https://www.nltk.org/api/nltk.tokenize.html

[4] https://www.nltk.org/api/nltk.tag.html

[5] `https://metamap.nlm.nih.gov/`

[6] `http://bioinformatics.ua.pt/becas/`

recongizers (NERs) which can annotate biomedical text with concepts that are included in biomedical structured resources. MetaMap was developed at the National Library of Medicine and its purpose is to map biomedical text to the Metathesaurus of the Unified Medical Language System using knowledge intensive approach based on symbolic NLP and computational linguistic techniques. On the other hand, BeCAS is an API for biomedical concept identification that works by combining the ability to select multiple concept types, reference external databases and automatically annotate nested and intercepted concepts. We use both BeCAS and MetaMap due to the fact that each one can locate different terms, consequently, there will be more candidate answers coverage. We take the union of the candidates extracted by each tool.

Figure 2 presents four questions along with their correct answers. We observe that all of the above techniques can help in extracting the correct answer. In question (d), *UUCCUUAAC* was not identified as a biomedical term, but it was identified and extracted as being a noun. BeCAS and MetaMap locate different terms. In question (a), MetaMap identifies *proprotein convertase subtilisin/kexin type 9* as a term, while BeCAS identifies the term *subtilisin*, as well as the term *kexin*. Similarly, in question (b) BeCAS recognizes the term *S-adenosyl-L-methionine*, while MetaMap identifies only *methionine*. Finally, in question (c), BeCAS identifies the term *alpha-galactosidase A*, while MetaMap cannot capture the *A*.

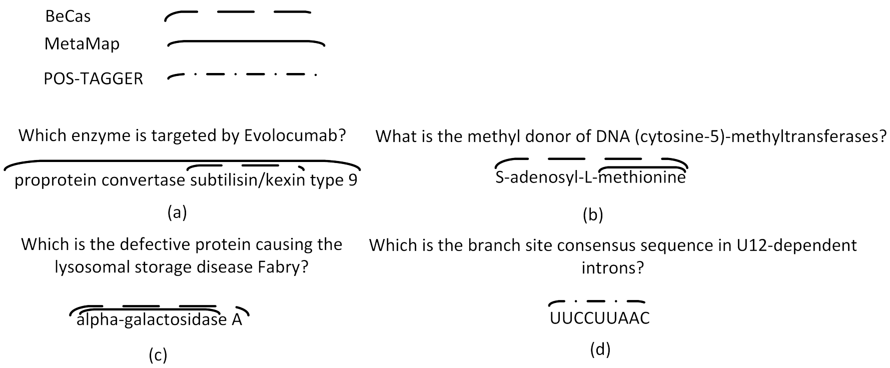

BeCas
MetaMap
POS-TAGGER
Which enzyme is targeted by Evolocumab?
proprotein convertase subtilisin/kexin type 9
(a)
What is the methyl donor of DNA (cytosine-5)-methyltransferases?
S-adenosyl-L-methionine
(b)
Which is the defective protein causing the lysosomal storage disease Fabry?
alpha-galactosidase A
(c)
Which is the branch site consensus sequence in U12-dependent introns?
UUCCUUAAC
(d)

Figure 2: The effect of MetaMap, BeCAS and POS-TAGGER in answer extraction phase.

3.2. Answer Representation

The answer representation phase converts the extracted candidate answers to multi-dimensional feature vectors. We firstly analyze the question to extract important information for answer representation, such as question elements and the question's type. Then, we extract the following three classes of features based on:

1. Textual Sources (TS), i.e. the question and passage.
2. Semantic Knowledge (SK), i.e. external resources with semantic annotation capability.
3. Word Embeddings (WE).

3.2.1. Identifying Question Elements and Question Types

The *lexical answer type* (LAT) of a question is the element revealing what the question actually asks for [36]. For example, the LAT of the question *"Which enzyme does MLN4924 inhibit?"* is the word *enzyme*. To identify the LAT of a question, we use two simple pattern based extraction methods [8]. In the first one, LAT falls into the first noun phrase (NP) of a question after the what/which. In the second one, LAT falls into the second NP of the pattern "NP[WHAT | WHICH]VP[BE]NP[*]". These two patterns are very successful in identifying the LAT, when it is included in the question, as we see later in the results section. The patterns fail to identify the LAT in only a few questions, such as "Willis-Ekbom disease is also known as?".

Question properties are question elements that offer additional information about the correct answer [37]. For example, the Question property of the question *"Which enzyme does MLN4924 inhibit?"* is the word *MLN4924* and offers the additional information of being a substance that inhibits the enzyme in question. To extract the properties of a question, we input the question body to MetaMap, BeCAS. The final set of question properties consists of all identified terms excluding the LAT.

Information about the correct answer of a question can be also derived from the structure of the question. Different structures indicate different question types [38]. Consider for example the following questions:

1. Is the transcriptional regulator BACH1 an activator or a repressor?
2. How many genes are imprinted in the human genome?
3. Where in the cell do the proteins S100A4 and p53 interact?

The answer to the first question can be found in the body of the question and it is an *activator* and/or a *repressor*. The second question expects a quantity as an answer: *fewer than 100*. The answer to the last question can be found in a prepositional phrase: *in the nucleus* or *in the cell nucleus*.

We define the following four question types, which are the most common ones found in the datasets that we experimented with: CHOICE, QUANTITY, LOCATION, GENERAL. To classify questions by type, we use simple pattern based extraction methods. In particular, CHOICE questions start with the verb *to be* and contain a conjunction word. QUANTITY questions start with the phrase *How many* or *How much*. LOCATION questions start with *where*. All the other questions are classified as GENERAL.

3.2.2. Features Based on Textual Sources

Prominence: This feature corresponds to the relative frequency of appearance of candidate answer, c, in the set of sentences, S, of the retrieved passages:

$$pr(c) = \frac{\sum_{s \in S} I(c \in s)}{|S|}$$

where I is the indicator function that returns 1 if candidate answer c is contained in sentence s and 0 otherwise. Intuitively, we expect correct answers to have a high prominence score. The assumption is based on the fact that the sentences are strongly related with the question. These sentences contain words of the question but also contain some terms that could be considered potential candidate answers. If a term is included in more than one sentences, then it is more probable to be the correct answer.

Weighted Prominence: Some of the sentences of the passages match better with the question than others. To represent this aspect, we compute three features that weight the frequency of a candidate answer, c, with the cosine similarity, Levenshtein distance and fuzzy similarity respectively between the question and each sentence, s, in the the set of sentences, S, of the retrieved passages:

$$wpr_i(c) = \frac{\sum_{s \in S} sim_i(q, s) * I(c \in s)}{\sum_{s \in S} sim_i(q, s)}$$

where i $\in$ {cosine, Levenshtein, fuzzy}.

We use the above three measures to estimate the distance between the question and a sentence due to the fact that each measure computes this distance from a different aspect. Levenshtein distance [39] computes the cost of the least expensive set of insertions, deletions or substitutions that would be needed to transform one string into the other. Fuzzy similarity estimates the matching between two strings without counting the order of the words inside them[7].

Number of Words in Candidate Answers: This feature corresponds to the length of the answer in words. Some answers need more details than others, consequently, these answers contain more words. Using this feature, we inform our model about the length of a candidate answer.

Co-existence Score: This boolean feature checks whether the candidate answer co-exists with the LAT in a sentence.

[7] https://github.com/seatgeek/fuzzywuzzy

$$co\_existence(c, LAT) = \begin{cases} 1, & \text{if c and LAT} \in \text{s.} \\ 0, & \text{otherwise.} \end{cases}$$

We expect that the correct answer will be close enough to the LAT of the question.

Question Type: This feature takes values from 0 to 3 according to the type of the question. Using this feature, we inform the model about structure of the question. We expect that the questions of different question types will be treated differently from the learning model.

Role Feature: This boolean feature checks whether a candidate answer is part of the question. This feature is uefull for the questions of CHOICE. We expect that if the candidate answer is part of the CHOICE question, then it is probable to be the correct answer. On the other hand, if the question is not a CHOICE question, then the candidate answer should not be included in the question.

3.2.3. Features Based on Semantic Knowledge

Wordnet: We use WordNet[8], a large lexical database of English, to extract the synonyms of a candidate answer, assuming that the fewer the synonyms the higher the chances of a correct answer. WordNet returns synonyms for single words, however, a candidate answer can consist of more than one words. Therefore, we define 3 features by considering the maximum, minimum and average number of synonyms respectively across all words in the candidate answer.

[8] https://wordnet.princeton.edu/

Type Coercion: This boolean feature checks whether the semantic type of the answer aligns with LAT's semantic type.

$$type\_coercion = \begin{cases} 1, & \text{if sem\_types(c)} \cap \text{sem\_types(LAT)} \neq \oslash \text{ .} \\ 0, & \text{otherwise.} \end{cases}$$

To identify the semantic types of these elements, we utilize MetaMap and BeCAS, which annotate the elements and extract a list of terms along with their semantic types. The semantic types of textual elements is the unification of the semantic types produced by MetaMap and BeCAS.

3.2.4. Features based on Word Embeddings

Each textual source (i.e. question, sentence, candidate answer, LAT) is a sequence of tokens. We convert each token to a multi-dimensional feature vector using the Word2Vec framework. To represent the whole source, we make the assumption that a source is the centroid of all its tokens vectors. This is a very common approach for building representations for longer phrases from single word vectors [40]. Additionally, we use cosine to estimate the similarity between two centroids. The assumption is that the cosine can estimate the similarity between two vectors in the vector space, consequently, we expect that if we encode question elements and candidate answers as vectors, the cosine could also estimate the similarity between them. Below, we present the features, we used, in detail.

Centroids: We create question, answer and LAT vectors. These vectors are used as features.

Cosine Similarity Between Centroids: we extract three features based on cosine similarity between (1) LAT and candidate answer (2) question and candidate answer (3) question property and candidate answer. Due to the fact that a question could have more than one properties, we compare the candidate answer vector with each of them and we select the property with the highest cosine similarity score.

3.3. Answer Ranking

Answer ranking is accomplished by learning a binary classifier that can output a probability or score with respect to the positive class. In Section 5 we present experimental results using three such learning algorithms: logistic regression (LR) [41], support vector machines (SVMs) [42] and extreme gradient boosting (XGBoost) [43]. Given a training set of questions, the set of correct answers per question, and a set of candidate answers per question, extracted by our approach, we construct a binary training set as follows. Each candidate answer of each question becomes a training example represented with the input features discussed in Section 3.2. Candidate answers that exist in the set of correct answers of their question are considered as positive examples, while the rest of the candidate answers are considered negative examples.

Given a new question and related passages, our approach will first extract a set of candidate answers, it will represent each of them as a feature vector and it will give them as input to the trained binary classifier. The classifier will output a score, indicative of the correctness of each candidate answer. Our system shall then output a ranked list of all candidates according to this score.

4. Experimental Setup

We describe the dataset that we used to train and test our approach, as well as the available measures and the learning process.

4.1. Data

BioASQ is an annual challenge in large-scale biomedical semantic indexing and question answering, running since 2013 [7]. The challenge comprises two main tasks: a) large-scale semantic indexing, and b) question answering. The question answering task further comprises two phases. The first one is focused on information retrieval. Given a question, participants must respond with a set of related documents, passages, RDF triplets, and concepts. In the second phase, for each question participants are also given the gold (correct) documents and passages containing the answer and must respond with exact answers and/or ideal answers, meaning whole paragraphs summarizing the most relevant information from the given passages. Questions fall into one of the following four types: list, factoid, summary, yes/no. We focus on the second phase of this challenge on factoid questions.

BioASQ organizers provide the participants with a dataset of questions, related documents and passages along with the corresponding answers[9]. 619 questions out of 2251 questions in this dataset are factoid questions. We also

[9] http://participants-area.bioasq.org/general_information/Task6b/

participated to BioASQ 2018 and our approach was evaluated on 5 test batches. Particularly, each test set contains 100 questions. The test set from the first test batch contains 31 factoid questions, the second one contains 21, the third one contains 32, the forth one contains 33 and the last one 44.

Due to the method that we used to extract candidate answers (i.e. nouns, biomedical terms) our approach can only answer 307 of the 619 questions. We decided to experiment both for keeping all questions in training set and keeping only the questions that can be answered. Making these experiments, we will show: (a) the overall performance of our approach in the dataset of 619 questions (b) the performance of our approach ignoring the negative results of answer extraction phase in the dataset of 307 questions (c) the significance of features in learning process (d) the performance of the answer extraction phase using several annotators in the dataset of 619 questions. From now on, we call the dataset with 619 questions as D1 and the dataset with 307 as D2.

We also use the dataset of word vectors provided by BioASQ[10] to create the centroid vectors for each textual source as described in section 3. The dataset contains word vectors for 1,701,632 distinct words extracted from a collection of 10,876,004 English abstracts of biomedical articles from PubMed. Particularly, they applied word2vec with the dimensionality of the vector space set to 200 using the continuous bag of words (CBOW) model [44].

[10] http://participantsarea.bioasq.org/tools/BioASQword2vec/

4.2. Measures

To evaluate our system we use the measures proposed by BioASQ for factoid QA. In detail,

1. **strict accuracy (Sacc)**: counts a question as correctly answered if the extracted answer is the first element of the returned list

   $$Sacc = c_1/n$$

   where n is the total number of questions and c_1 is the number of factoid questions that have been answered correctly when only the first element of each returned list is considered.

2. **lenient accuracy (Lacc)**: counts a question as correctly answered if the candidate answer is included, not necessary as the first element, in the returned list.

   $$Lacc = c_5/n$$

   where c_5 is the number of questions that have been answered correctly in the lenient sense.

3. **mean reciprocal rank (MRR)**: for each question q we search the returned list looking for the topmost position that contains the candidate answer. If the topmost position is the $j-th$ one then $r(i) = j$; otherwise $r(i) \to +\infty$, i.e., $1/r(i) = 0$

   $$MRR = \frac{\sum_{i=1}^{n} \frac{1}{r(i)}}{n}$$

4.3. Evaluation Process

We use k-fold cross validation where k depends on the size of dataset. The purpose of this validation is to split the dataset in k independent datasets where the size of each dataset will be equal to 10. The evaluation is finished after k iterations selecting in each iteration a different dataset as test set. Using this process, the training set is bigger and we can get more estimations for our approach. Two kinds of results, we present.

Firstly, we present results on the D1 and D2 using LR in order to show the impact of several annotators in answer extraction phase and the impact of different classes of features in answer ranking. We use LR because many participants use this machine learning algorithm in the learning phase and the obtained results were promising. Each produced learning model was evaluated using the k-fold cross validation. We also used paired t-test to estimate the significance of our results.

Next, we present results from our participation to the BioASQ Challenge using a voting scheme and tuning. Particularly, we focus on three algorithms (LR, SVMs and XGBoost) and we finally call a voting scheme. We tune each algorithm separately and we also tune the weights of the algorithms in the voting scheme. The produced learning models are evaluated based on MRR and the final learning model is chosen when the MRR has the highest score. As dataset, we used a subset of the D1. Particularly, we use the questions provided by BioASQ(2013-2015)(324 questions) keeping the questions which can be answered by our system (147 questions). For each model produced by changing its parameters, we applied k-fold cross validation.

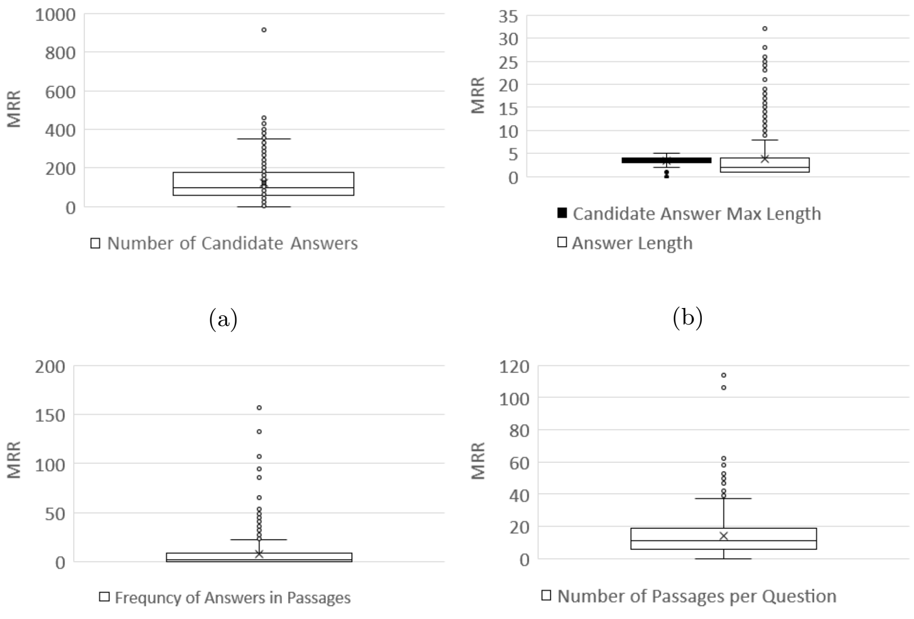
5. Results

In this section, we present results on both D1 and D2. Firstly, we mention some descriptive statistics in the D1 emphasizing on the absence of correct answers in the list of candidate answers. Next, we present results for the answer extraction phase using the D1. Particularly, we show the impact of different annotators in the extraction of candidate answers from passages. Afterwards, we report the results for answer representation phase using both the D1 and D2 and applying undersampling in D2. Making the latter experimentation, we will observe: (a) the effect of the noisy D1 dataset, (b) the effect of underasampling in the overall performance of our approach, (c) the most useful features in ranking phase and (d) the impact of word embeddings in ranking phase. Finally, we present the results of the tuning process and the results against the other participants in BioASQ Challenge (2017-2018).

5.1. Limitations in Our Approach

The AP is a difficult task in biomedical domain. We present some descriptive statistics of the training set that indicates the limitations, we have, in our approach.

Figure 3c presents the frequency of correct answers in the passages. To identify the correct answers in the passages, we seek an answer as normalized substring into normalized passages and we count the number of occurrences of substring in them. As an example, NCX and (NCX) and NCX1-heterozygous



(a)

(b)

(c)

(d)

Figure 3: Descriptive statistics on D1 dataset. (a) presents the number of candidate answers per question. (b) shows the length of the answers calculating the number of the included words comparing to the length of candidate answers extracted by our approach ,(c) presents the frequency of each correct answer in the passages for each question. (d) reports the number of passages that are available for each question

count 3 for the correct answer NCX. We count each occurrence of substring in passages despite the fact that some substrings concatenate with other words or symbols. The assumption here is that we can apply different methods in preprocessing phase to isolate the candidate answer. For instance, we could ignore the parenthesis in (NCX) in order to keep the abbreviation NCX as candidate answer. Furthermore, some questions have more than one answers, however, we count the frequency of all and we assume that this frequency corresponds to one answer. The main observations are: (a) most of the correct answers (58%) are very rare. Indeed, there are 139 answers where the frequency is one or two and the rest 206 answers do not exist exactly in passages. Consequently, the approaches that extract candidate answers only from passages, can achieve at most 67% accuracy in this dataset (b) 11% of answers occur more than 20 times in the passages. Approaches that rank candidate answers based on term frequency could achieve good results on the 11% of the questions.

The distribution of answers frequency in the dataset indicates the limitations of current approaches in answer extraction. An another limitation which is observed during our analysis, is the number of words in the correct answers. In Figure 3b, most of the correct answers contain at most 5 words. However, 19% of the questions contain more. Consequently, many approaches, including ours, can not answer these questions correctly (our answer extraction method can not identify answers that contain more than 4 words).

These limitations are crucial in our AP approach. Nevertheless, the performance of our approach is not only affected by the dataset, it is also affected by the extraction of candidate answers without applying any kind of filtering and the multiple annotators, we use. After the execution of the answer extraction phase, 76920 candidate answers were produced. Figure 3a shows the number of candidate answers per question. The high number of candidate answers is justified due to the fact that there are many passages for some questions (Figure 3d). The number of correct answers is 730 in the dataset (619 plus the answers variants contained in dataset). Consequently, the 99.1% of the candidate answers are not the expected correct answers. Mentioning the analysis, we made previously, this rate is bigger. However, as we will see later, the number of annotators improve the performance of the AP despite the creation of an imbalance dataset. Furthermore, although the limitations are significant, our approach is competitive against other state of the art approaches in this task.

5.2. Results on Training Sets

We experimented with the identification of LAT in the D1 dataset. We observed that using the two patterns described earlier, we can identify the LAT in 88% of the questions. Particularly, 47% of the questions fall into the first pattern and the rest 41% fall into the second one. The rest 12% of the questions either do not include LAT or do not fall into the two proposed patterns. In this 12% of the questions, 3% are QUANTITY questions, 2% LOCATION questions and 2% CHOICE questions. The rest 4% of the questions can not be interpreted by our system correctly.

In the answer extraction phase, we proposed several annotators to identify candidate answers resulting in a very large amount of candidate answers. Thus, it is necessary to evaluate our answer extraction method in terms of the overall performance of the answer representation. Table 1 indicates the impact of the annotators to the overall performance of the AP. We observe that using several annotators, the performance is better despite the increment of wrong answers. Furthermore, part of speech tagging increases the performance when it is used together with biomedical annotators. Finally, BeCAS can answer more questions than MetaMap, though, the results are worse in terms of MRR and Sacc.

To the best of our knowledge, BeCAS has not already been used in biomedical answer extraction, consequently, we applied statistical tests to estimate the impact of using this tool. The results of the tests are included in Table 1.

Annotators	D1 Dataset			D2 Dataset			
	MRR	Lacc	Sacc	MRR	Lacc	Sacc	#Questions
MetaMap + BeCAS + POS-TAGGER	.2553	.3639*	.2033	.5149	.7500*	.4167	307
MetaMap + POS-TAGGER	.2509	.3410	.2082	.5041	.6840	.4165	273
BeCAS + POS-TAGGER	.2360*	.3459*	.1836	.4758*	.6974*	.3702	280
MetaMap + BeCAS	.2218	.3262*	.1656	.4472	.6577*	.3301	273
MetaMap	.2067	.2787	.1672	.4103	.5618	.3371	217
POS-TAGGER	.2063	.2590	.1872	.4143	.5222	.3709	182
BeCAS	.1902	.2738	.1475	.3835	.5521	.2974	220

* significant at $p < 0.05$.

Table 1: Results on data sets D1 and D2, sorted by MRR, using different combinations of annotators. The last column presents the number of questions for which the correct answer was extracted. Statistical significance is estimated between pairs of models with and without the BeCAS tool, i.e. line 1 vs line 2, line 3 vs line 6 and line 4 vs line 5.

Despite that nouns add noise (non-biomedical terms) to our dataset, we observe that the best results in all measures (MRR, Lacc, Sacc) are obtained when using both biomedical terms and nouns. Intuitively, we can explain this situation making reference to the performance of the biomedical NERs. Some of the correct answers which are named entities, can not be identified as biomedical terms by NERs. For instance, some terms are not included in the vocabularies of the NERs, however, these terms could be considered as potentially correct answers. On the other hand, the most probable word class for the one-word expected answer is noun. Consequently, if a term is not included in the vocabularies of the biomedical tools, is probable to be considered as noun. Thus, using both nouns and terms, we can achieve better performance in answer extraction.

Classes Of Features	D1 Dataset			D2 Dataset		
	MRR	Lacc	Sacc	MRR	Lacc	Sacc
TS + SK + WE	.2553	.3639	.2033	.5149	.7500	.4167
TS + WE	.2546	.3656	.1951	.5163*	.7400*	.4100
TS +SK	.2376	.3508	.1721	.4839	.7167	.3567
TS	.2243	.3377	.1639	.4557	.6833	.3367
WE + SK	.1265*	.2328*	.0705*	.2588*	.4800*	.1500*
WE	.1201	.2213	.0689	.2511	.4667	.1467
SK	.0887	.1574	.0525	.2381	.4333	.1333

* significant at $p < 0.05$.

Table 2: Results on classes of features in datasets D1 and D2. The statistical significance is estimated between models that use word embeddings with models that do not use them.

To evaluate the utility of word embeddings in answer representation phase, we present coarse-grained results. Coarse-grained results come from the 3 classes of features mentioned in section 3 (1) Textual Sources (TS) (2) Semantic Knowledge (SK) and (3) Word Embeddings (WE). In table 2 we observe that word embeddings improve the results in all cases. Furthermore, features based on TS are significant for system's accuracy. Combining both of three classes we achieve the highest scores. The impact of SK is greater in D1 dataset rather than in D2 when SK is combined with the other two classes of features.

To investigate more the impact of word embeddings in AP, we compared the returned list of candidate answers ranked using word embeddings and those candidate answers ranked without word embeddings. We observe an interesting feature of word embeddings. They capture lexical semantics and have been shown that are effective in word analogy task [45]. Although, we use them in a different task (i.e. answer processing), we observe that they capture lexical semantics in the list of candidate answers. For instance, the answer of a question *"Inhibition of which transporter is the mechanism of action of drug Canagliflozin?"* is the *"sodium glucose co-transporter 2"*. The top-5 elements in the list of candidate answers after ranking phase are *"diabetes melitus"*, *"SGLT2"*, *"Sodium glucose co-transporter 2"*, *"sodium-glucose co-transporter 2"* and *"sodium glucose contraporter 2"*. The candidate answers except the first one, are lexical variants. In the most cases, when the correct answer has lexical variants, we observe the same pattern. On the other hand, without using word embeddings the answers of the same question are *"SLGT2"*, *"inhibitors"*,*"inhibitor"*,*"sodium"* and *"type 2 diabetes mellitus"*.

Thus, we believe that word embeddings improve the performance of AP because they capture similar candidate answers. On the other hand, without using word embeddings, we may capture the correct answer, however, the returned answer could be a lexical variant of the expected correct answer. For instance the answer *SLGT2* could be considered correct answer, however, in a string matching evaluation, it is considered wrong against the *sodium glucose* *co-transporter 2.*

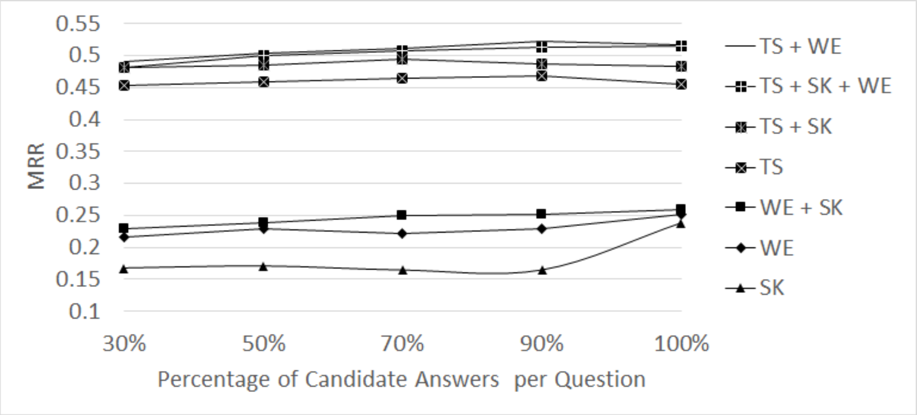
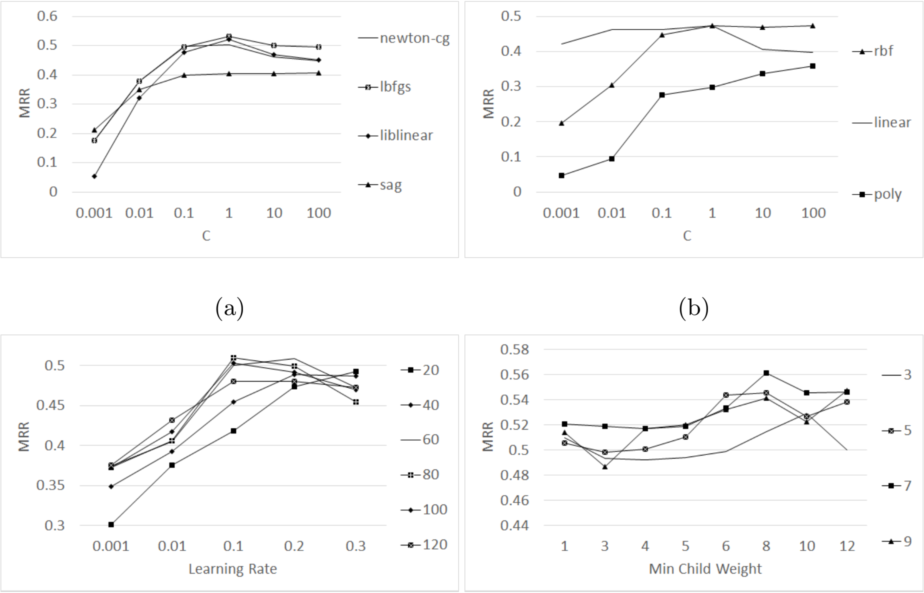


Figure 4: The effect of removing candidate answers in the performance of AP using different classes of features in the D2 dataset

As we described earlier, answer extraction technique produces a big list of candidate answers in which the number of correct one is limited. Consequently, the dataset which is used in answer ranking phase contains much more negative instances than positive. Due to this extreme class imbalance issue, we expect that the system performance is affected. To address this problem, we applied an under-sampling method in order to achieve a more balanced class distribution [46]. Particularly, we experimented on removing candidate answers from the D2 using k-fold validation. In the training set, we randomly removed 10%, 30%, 50% and 70% of candidate answers per question. On the other hand, the test set was not changed. We also experimented using different classes of features. Figure 4 reports results applying under-sampling. We observe that using 10% less candidate answers per question, the results are quite better. Reducing more candidate answers, the results are worse. However, the number of negative examples do not significantly affect the performance of the AP.

5.3. Comparing with other Systems

In our participation to the BioASQ Challenge, we called ensemble methods and we also tuned their parameters.



(a)

(b)

(c)

(d)

Figure 5: The tuning process on Logistic Regression, SVMs and eXtreme Gradient Boosting.

In detail, we tuned the C parameter that estimates the inverse of regularization strength both for LR (Figure 5a) and SVM (Figure 5b) (Table 3). The smaller the value of C, the stronger the regularization is. Furthermore, we tested on different LR solvers and different kernels for SVM. For XGBoost algorithm, we tuned the learning rate that determines the effect of each tree on the final result and the number of estimators (Figure 5c)(Table 4). We also parametrized the boosting parameters, min child weight and maximum depth of a tree (Figure 5d)(Table 5). The default estimated parameters defined in the scikit-learn[11] package in python for SVM and LR. For the default XGBoost parameters details provided in the website[12]. Finally, we used the above three tuned algorithms in a voting scheme that is also tuned on the weights of each algorithm in the final model.

C parameter	SVMs			LR			
	rbf	linear	poly	newton-cg	lbfgs	liblinear	sag
0.001	.1969	.4210	.0464	.1767	.1767	.0523	.213
0.01	.3040	.4632	.0946	.3776	.3776	.3213	.3495
0.1	.4485	.4626	.2762	.4973	.4962	.4780	.3994
1	.4739	.4729	.2980	.5033	.5315	.5205	.4042
10	.4700	.4063	.3373	.4615	.4990	.4688	.4045
100	.4730	.3969	.3590	.4494	.4956	.4504	.4055

Table 3: MRR score for each algorithm tuning the C parameter along with the solvers for LR and kernels for SVMs.

We summarize our observations as follows: (1) the lbfgs solver with $C = 1$ indicates the highest MRR (approximately 53%) against the SVMs and the other solvers of LR. (2) The liblinear solver is close enough to lbfgs (52% MRR) (3)

[11] http://scikit-learn.org/

[12] http://xgboost.readthedocs.io

#estimators	Learning Rate				
	0.001	0.01	0.1	0.2	0.3
20	.3015	.3758	.4182	.4731	.4924
40	.3485	.393	.454	.4885	.4868
60	.3744	.4049	.4998	.5089	.4725
80	.3726	.4058	.5101	.4993	.4548
100	.3726	.4177	.5027	.4914	.4698
120	.3756	.4313	.4798	.4806	.4723

Table 4: MRR score of XGBoost for the pair of parameters Learning Rate and number of estimators

The poly kernel seems to increase the results for greater values of C, however, the highest MRR is approximately 36%. (4) XGBoost indicates better results when learning rate is equal to 0.1 (default value) and the number of estimators is 80 (51% MRR score). Giving the most appropriate combination of estimators and learning rate based on their values (i.e. 80 and 0.1 respectively), we tuned the min child weight and maximum depth of a tree. We observe that using maximum depth of a tree equals to 7, for each min child weight the MRR is better in most of cases. The highest MRR 56% is observed when maximum depth is equal to 7 and the min child weight is 8. (5) Giving greater weight to XGBoost, the voting scheme achieves the highest MRR 58% (Table 6).

LR models are affected by the existence of outliers compared to SVM models.

Min Child Weight	Maximum Depth			
	3	5	7	9
1	.5101	.5055	.5206	.5138
3	.4936	.4983	.5187	.4865
4	.4923	.5008	.5173	.5169
5	.4939	.5102	.5187	.52
6	.4986	.5438	.5333	.5323
8	.5146	.5454	.5614	.5411
10	.5293	.527	.5456	.5226
12	.4999	.5382	.5461	.5474

Table 5: MRR score of XGBoost for the pair of parameters min child weight with maximum depth of a tree.

However, we observe that most of the generalized linear models produced by LR using different solvers achieve better results than SVMs which are not affected by outliers. Furthermore, the non-linear kernel Radial Basis Function (RBF) is controlled by the C parameter, however, the results are still worse than linear models. We believe that the improved results of LR are due to the probabilistic outputs produced by the LR models. The LR models have been used for this purpose as we described in literature review section and the obtained results were promising. As XGBoost is an ensemble method, its improved performance compared to the single models of LR and SVM was an expected result.

Voting Scheme Tuning					
Weights			MRR	Lacc	Sacc
SVMs	LR	XGBoost			
1	1	1	.5568	.7357	.4500
2	1	1	.5627	.7500	.4500
2	1	2	.5768	.7500	.4643
2	2	1	.5529	.7286	.4500
1	1	2	**.5826***	**.7643***	**.4786***
1	2	1	.5614	.7357	.4571

* significant at $p < 0.05$.

Table 6: A voting scheme using SVMs, LR and XGBoost with their weights. The statistical significance is estimated between the best model with all the other models.

We submitted results from 3 variations of our approach. The *fa1* system is our basic approach as described in section 3. The *fa2* system removes some candidate answers that it couldn't be considered as correct answers after the ranking phase. Particularly, we remove answers that have more than one synonyms based on WordNet.

Both of two systems use the training set described above. The model is tuned as described earlier. The *fa3* system is our basic approach using the D2. We also used the same parameters in the learning process as in *fa1* and *fa2* systems.

Due to the fact that D2 dataset contains questions from the BioASQ 2017, we do not use the *fa3* system in the comparison with the participants of BioASQ 2017. Furthermore, because the automated evaluation can miss some correct answers, we also present the results of *fa1* which are produced by manual evaluation. The system's answers, we assume that is correct, are presented in Appendix.

Table 7 summarizes the results of our approach against the top approaches in BioASQ 2017. The quality of the systems is evaluated by the MRR score. The results of our approach are better comparing with the BioASQ baseline and competitive against the top 2 systems. Lab Zhu, Fudan system [11] uses PubTator to extract the candidate answers and term frequency to rank them. We observe that this approach can beat the neural network based approach (Deep QA) in test batches 2 and 5 using a simpler approach than Deep QA. However, we overcame this approach in batch 1. We expect that Lab Zhu, Fudan system will be very limited in real QA applications because as we mentioned in the limitations of our approach section, the frequency of the correct answer in passages is very low in a larger dataset which contain questions from 5 years of BioASQ Challenge. On the other hand, Deep QA [18] is based on current state of the art approaches which use word embeddings to encode the input textual sources and a neural network model to predict the boundaries of the answer into the passages. However, the performance of Deep QA system doesn't explicitly depend on the neural model, but, it also depends on the assumptions that the authors made. They proposed a context/type matching heuristic technique as a backbone of their approach. The proposed rules are strongly related with the dataset. These rules can identify fewer candidate answers than our system, but, these candidate answers are more probable to be the correct answers. However, due to the fact that the rules were proposed to fit on the provided dataset, we do not expect that this approach will also be the state of the art for other datasets. Finally, after the manual evaluation of our approach, we observed that our system can overcome the Deep QA in the 5th test batch.

Table 8 summarizes the results of our approach against the BioASQ 2018 participants. We observe that the AP Task in biomedical domain is a difficult task. The highest scores, MRR, Lacc and Sacc are approximately 43%, 62% and 33% respectively. Our approach overcomes the others in 3/5 test batches. Furthermore, we achieve the highest results based on Lacc. On the other hand, using Sacc, our approach is worse than the system 2 in 4/5 test batches. System 3 has higher Sacc score in test batch 2. Systems 4 and 5 are lower in the ranking.

Finally, our approach overcomes the baseline system in all test batches.

ID	Systems	Batch 1			Batch 2			Batch 3			Batch 4			Batch 5		
		Sacc	Lacc	MRR	Sacc	Lacc	MRR	Sacc	Lacc	MRR	Sacc	Lacc	MRR	Sacc	Lacc	MRR
1	fa1	.4000	.5600	.4467	.2258	.4194	.2937	.2092	.4615	.3333	.2121	.3333	.2576	.2286	.4857	.3286
	fa2	.3600	.5200	.4333	.2258	.4194	.3038	.2692	.4615	.3462	.2424	.3333	.2828	.2286	.4857	.3271
	fa1**	.4400	.6800	.5352	.2903	.5161	.3071	.3461	.5769	.4165	.2727	.4242	.3181	.3714	**.5714**	**.4741**
2	Deep QA*	.5600	.6800	.6033	.3871	.5161	.4419	.3077	.5769	.4308	.3333	.5455	.4162	.3714	.4571	.3924
3	Lab Zhu, Fudan*	.4000	.4400	.4200	.4516	.5161	.4839	.3462	.4231	.3846	.2727	.4545	.3510	.4000	.5143	.4524
4	BioASQ Baseline*	.2800	.4000	.3333	.1613	.3548	.2215	.1154	.2692	.1923	.0303	.1212	.0682	.0571	.2000	.1167

*The best systems of the participants

**The fa1 system which is manually evaluated

Table 7: Comparing our system with the top 2 systems and the baseline of BiOASQ 2017. Bold scores indicates better performance than participants in a test batch.

ID	Systems	Batch 1			Batch 2			Batch 3			Batch 4			Batch 5		
		MRR	Lacc	Sacc	MRR	Lacc	Sacc	MRR	Lacc	Sacc	MRR	Lacc	Sacc	MRR	Lacc	Sacc
1	fa1	.2376	.3226	.1935	.3468	.5238	.2381	**.2604**	**.4063**	.1563	.2374	**.2727**	.2121	.1758	**.2955**	.1136
	fa2	**.2484**	**.3548**	.1935	.3746	**.6190**	.2381	.2135	.3125	.1563	**.2475**	**.3030**	.2121	.1136	.2727	.1758
	fa3	.2376	.3226	.1935	.3270	.5238	.2381	**.3083**	**.4375**	**.2500**	.2283	**.2727**	.2121	.1136	**.2955**	.1777
2	Lab Zhu,Fudan*	.2419	.2581	.2258	.4325	.5714	.3333	.2370	.3125	.1875	.2424	.2424	.2424	.2273	.2727	.2045
3	OAQA*	.2366	.3226	.1613	.2857	.2857	.2857	.2094	.3438	.1250	.1313	.1212	.0909	.1951	.2273	.1818
4	YODAQA*	.0484	.0968	.0000	.1429	.1905	.0952	.1094	.1563	.0625	-	-	-	-	-	-
5	SpanBaseline*	.0968	.0968	.0000	-	-	-	-	-	-	-	-	-	-	-	-
6	BioASQ Baseline	.2403	.2903	.2258	.1841	.2857	.1429	.2396	.2813	.2188	.0859	.1212	.0606	.0795	.1818	.0000

*The best systems of the participants

Table 8: Comparing our system with the best systems of BiOASQ 2018. Bold scores indicates better performance than participants in a test batch.

6. Conclusions and Future Work

This work addressed the challenge of AP in the context of biomedical QA. We showed that despite the increasing number of candidate answers, produced by several annotators, and the class imballance issue do not influence the performance of answer processing. In addition, we introduced a novel answer representation technique based on word embeddings and external resources. Finally, we experimented with several supervised learning algorithms to build a learning model for ranking the candidate answers.

In the answer extraction phase, we assumed that a candidate answer is a a noun, a biomedical term or a number. This assumption affects the system's performance. For further work, we can extend the list of candidate answers by employing various answer forms (e.g. noun phrases, prepositional phrases, bigrams, trigrams etc.). The bigger the set of candidate answers, the more probable to contain the correct answer.

Simple answer extraction techniques entail complex answer ranking methods. The type coercion score determines whether the candidate answer satisfies the answer-type requirements of the question. Nevertheless, this scoring function returns a value only if the semantic type of the candidate answer aligns with the semantic type of LAT. We can change this score to compute the similarity between semantic types [47]. Thus, we can estimate the semantic distance between the answer and the answer type. Furthermore, we observed that word embeddings can improve the accuracy of an AP model, despite the use of a simple technique for the conversion of question elements, sentences, and candidate answers to vectors. In the future, we can use embeddings produced by other frameworks (e.g. GloVe [48]) or embeddings that focus on sentence level rather than word level.

Semantic Role Labeling (SRL) can be used both in answer extraction and answer ranking phase. Particularly, in answer extraction phase, we can extract answers based on how well the candidate answers match the predicted answer type. This approach matters only if the the system's accuracy is not affected by incorrect matches. Furthermore, in answer ranking phase, we can use an SRL tool, like BioSMILE [49] to generate semantic features, as described in [50].

Acknowledgements

This work was partially funded by Atypon Systems Inc[13]. We would like to express our very great appreciation to the BioASQ for the provided datasets in biomedical question answering.

References

[1] O. Kolomiyets, M. Moens, A survey on question answering technology from an information retrieval perspective, Inf. Sci. 181 (24) (2011) 5412–5434. doi:10.1016/j.ins.2011.07.047.
URL https://doi.org/10.1016/j.ins.2011.07.047

[2] H. T. Dang, D. Kelly, J. J. Lin, Overview of the TREC 2007 question an-

[13] https://www.atypon.com/

swering track, in: Proceedings of The Sixteenth Text REtrieval Conference, TREC 2007, Gaithersburg, Maryland, USA, November 5-9, 2007, 2007.
URL http://trec.nist.gov/pubs/trec16/papers/QA.OVERVIEW16.pdf

[3] S. J. Athenikos, H. Han, Biomedical question answering: A survey, Computer Methods and Programs in Biomedicine 99 (1) (2010) 1–24. doi:10.1016/j.cmpb.2009.10.003.
URL https://doi.org/10.1016/j.cmpb.2009.10.003

[4] D. Jurafsky, J. H. Martin, Speech and Language Processing, 3rd Edition, Unpublished Draft, 2017, Ch. 28.

[5] E. Brill, S. Dumais, M. Banko, An analysis of the askmsr question-answering system, in: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, Association for Computational Linguistics, 2002, pp. 257–264.

[6] M. Paşca, Open-domain question answering from large text collections (2003).

[7] G. Tsatsaronis, G. Balikas, P. Malakasiotis, I. Partalas, M. Zschunke, M. R. Alvers, D. Weissenborn, A. Krithara, S. Petridis, D. Polychronopoulos, Y. Almirantis, J. Pavlopoulos, N. Baskiotis, P. Gallinari, T. Artières, A. N. Ngomo, N. Heino, É. Gaussier, L. Barrio-Alvers, M. Schroeder, I. Androutsopoulos, G. Paliouras, An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition, BMC Bioinformat-

ics 16 (2015) 138:1–138:28. doi:10.1186/s12859-015-0564-6.
URL https://doi.org/10.1186/s12859-015-0564-6

[8] D. Weissenborn, G. Tsatsaronis, M. Schroeder, Answering factoid questions in the biomedical domain, in: Proceedings of the first Workshop on Bio-Medical Semantic Indexing and Question Answering, a Post-Conference Workshop of Conference and Labs of the Evaluation Forum 2013 (CLEF 2013) , Valencia, Spain, September 27th, 2013., 2013.
URL http://ceur-ws.org/Vol-1094/bioasq2013_submission_5.pdf

[9] M. Sarrouti, S. O. E. Alaoui, A biomedical question answering system in bioasq 2017, in: BioNLP 2017, Vancouver, Canada, August 4, 2017, 2017, pp. 296–301. doi:10.18653/v1/W17-2337.
URL https://doi.org/10.18653/v1/W17-2337

[10] Y. Papanikolaou, D. Dimitriadis, G. Tsoumakas, M. Laliotis, N. Markantonatos, I. P. Vlahavas, Ensemble approaches for large-scale multi-label classification and question answering in biomedicine, in: Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014., 2014, pp. 1348–1360.
URL http://ceur-ws.org/Vol-1180/CLEF2014wn-QA-PapanikolaouEt2014.pdf

[11] S. Peng, R. You, Z. Xie, B. Wang, Y. Zhang, S. Zhu, The fudan participation in the 2015 bioasq challenge: Large-scale biomedical semantic indexing and question answering, in: Working Notes of CLEF 2015 - Con-

ference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015., 2015.
URL http://ceur-ws.org/Vol-1391/88-CR.pdf

[12] B. Carpenter, Lingpipe for 99.99% recall of gene mentions, in: Proceedings of the Second BioCreative Challenge Evaluation Workshop, Vol. 23, 2007, pp. 307–309.

[13] Z. Yang, N. Gupta, X. Sun, D. Xu, C. Zhang, E. Nyberg, Learning to answer biomedical factoid & list questions: OAQA at bioasq 3b, in: Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015., 2015.
URL http://ceur-ws.org/Vol-1391/114-CR.pdf

[14] Z. Y. Y. Z. E. Nyberg, Learning to answer biomedical questions: Oaqa at bioasq 4b, ACL 2016 (2016) 23.

[15] Y. Mao, C. Wei, Z. Lu, NCBI at the 2014 bioasq challenge task: Large-scale biomedical semantic indexing and question answering, in: Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014., 2014, pp. 1319–1327.
URL http://ceur-ws.org/Vol-1180/CLEF2014wn-QA-MaoEt2014.pdf

[16] E. Papagiannopoulou, Y. Papanikolaou, D. Dimitriadis, S. Lagopoulos, G. Tsoumakas, M. Laliotis, N. Markantonatos, I. Vlahavas, Large-scale semantic indexing and question answering in biomedicine, in: Proceedings of the Fourth BioASQ workshop, 2016, pp. 50–54.

[17] H. Yenala, A. Kamineni, M. Shrivastava, M. K. Chinnakotla, IIITH at bioasq challange 2015 task 3b: Bio-medical question answering system, in: Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015., 2015.
URL `http://ceur-ws.org/Vol-1391/55-CR.pdf`

[18] D. Weissenborn, G. Wiese, L. Seiffe, Making neural QA as simple as possible but not simpler, in: Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017, 2017, pp. 271–280. `doi:10.18653/v1/K17-1028`.
URL `https://doi.org/10.18653/v1/K17-1028`

[19] P. Baudis, J. Sedivý, Biomedical question answering using the yodaqa system: Prototype notes, in: Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015., 2015.
URL `http://ceur-ws.org/Vol-1391/131-CR.pdf`

[20] P. Baudiš, Yodaqa: a modular question answering system pipeline, in: POSTER 2015-19th International Student Conference on Electrical Engineering, 2015, pp. 1156–1165.

[21] G. Wiese, D. Weissenborn, M. L. Neves, Neural question answering at bioasq 5b, in: BioNLP 2017, Vancouver, Canada, August 4, 2017, 2017, pp. 76–79. `doi:10.18653/v1/W17-2309`.
URL `https://doi.org/10.18653/v1/W17-2309`

[22] D. Weissenborn, G. Wiese, L. Seiffe, Fastqa: A simple and efficient neural architecture for question answering, CoRR abs/1703.04816. `arXiv:1703.04816`.

URL `http://arxiv.org/abs/1703.04816`

[23] H. T. Ng, L. H. Teo, J. L. P. Kwan, A machine learning approach to answering questions for reading comprehension tests, in: Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13, Association for Computational Linguistics, 2000, pp. 124–132.

[24] X. Hao, X. Chang, K. Liu, A rule-based chinese question answering system for reading comprehension tests, in: Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007. Third International Conference on, Vol. 2, IEEE, 2007, pp. 325–329.

[25] Y. Du, H. Meng, X. Huang, L. Wu, The use of metadata, web-derived answer patterns and passage context to improve reading comprehension performance, in: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2005, pp. 604–611.

[26] K. Xu, H. Meng, Using verb dependency matching in a reading comprehension system, in: Asia Information Retrieval Symposium, Springer, 2004, pp. 190–201.

[27] W. Wang, N. Yang, F. Wei, B. Chang, M. Zhou, Gated self-matching networks for reading comprehension and question answering, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vol. 1, 2017, pp. 189–198.

[28] C. Xiong, V. Zhong, R. Socher, Dynamic coattention networks for question answering, arXiv preprint arXiv:1611.01604.

[29] S. Wang, J. Jiang, Machine comprehension using match-lstm and answer pointer, arXiv preprint arXiv:1608.07905.

[30] B. Pan, H. Li, Z. Zhao, B. Cao, D. Cai, X. He, Memen: multi-layer embedding with memory networks for machine comprehension, arXiv preprint arXiv:1707.09098.

[31] F. Wu, N. Lao, J. Blitzer, G. Yang, K. Weinberger, Fast reading comprehension with convnets, arXiv preprint arXiv:1711.04352.

[32] M. Hu, Y. Peng, Z. Huang, X. Qiu, F. Wei, M. Zhou, Reinforced mnemonic reader for machine reading comprehension, arXiv preprint arXiv:1705.02798.

[33] B. Dhingra, H. Liu, R. Salakhutdinov, W. W. Cohen, A comparative study of word embeddings for reading comprehension, arXiv preprint arXiv:1703.00993.

[34] A. R. Aronson, Effective mapping of biomedical text to the UMLS metathesaurus: the metamap program, in: AMIA 2001, American Medical

Informatics Association Annual Symposium, Washington, DC, USA, November 3-7, 2001, 2001.
URL http://knowledge.amia.org/amia-55142-a2001a-1.597057/t-001-1.599654/f-001-1.599655/a-003-1.600128/a-004-1.600125

[35] T. Nunes, D. Campos, S. Matos, J. L. Oliveira, Becas: biomedical concept recognition services and visualization, Bioinformatics 29 (15) (2013) 1915–1916. doi:10.1093/bioinformatics/btt317.
URL https://doi.org/10.1093/bioinformatics/btt317

[36] A. Lally, J. M. Prager, M. C. McCord, B. Boguraev, S. Patwardhan, J. Fan, P. Fodor, J. Chu-Carroll, Question analysis: How watson reads a clue, IBM Journal of Research and Development 56 (3) (2012) 2. doi:10.1147/JRD.2012.2184637.
URL https://doi.org/10.1147/JRD.2012.2184637

[37] K. Tymoshenko, A. Moschitti, Assessing the impact of syntactic and semantic structures for answer passages reranking, in: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015, 2015, pp. 1451–1460. doi:10.1145/2806416.2806490.
URL http://doi.acm.org/10.1145/2806416.2806490

[38] K. C. Litkowski, Question-answering using semantic relation triples, in: Proceedings of The Eighth Text REtrieval Conference, TREC 1999,

Gaithersburg, Maryland, USA, November 17-19, 1999, 1999.
URL http://trec.nist.gov/pubs/trec8/papers/clresearch.pdf

[39] W. J. Heeringa, Measuring dialect pronunciation differences using levenshtein distance, Ph.D. thesis, Citeseer (2004).

[40] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, A. Y. Ng, Grounded compositional semantics for finding and describing images with sentences, Transactions of the Association of Computational Linguistics 2 (1) (2014) 207–218.

[41] J. Friedman, T. Hastie, R. Tibshirani, The elements of statistical learning, Vol. 1, Springer series in statistics New York, NY, USA:, 2001.

[42] N. Cristianini, J. Shawe-Taylor, et al., An introduction to support vector machines and other kernel-based learning methods, Cambridge university press, 2000.

[43] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, 2016, pp. 785–794. doi:10.1145/2939672.2939785.
URL http://doi.acm.org/10.1145/2939672.2939785

[44] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781.

[45] T. Noraset, C. Liang, L. Birnbaum, D. Downey, Definition modeling:

Learning to define word embeddings in natural language, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA., 2017, pp. 3259–3266.
URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14827

[46] J. Prusa, T. M. Khoshgoftaar, D. J. Dittman, A. Napolitano, Using random undersampling to alleviate class imbalance on tweet sentiment data, in: Information Reuse and Integration (IRI), 2015 IEEE International Conference on, IEEE, 2015, pp. 197–202.

[47] J. J. Jiang, D. W. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy (1997) 19–33.
URL https://aclanthology.info/papers/O97-1002/o97-1002

[48] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, 2014, pp. 1532–1543.
URL http://aclweb.org/anthology/D/D14/D14-1162.pdf

[49] R. T. Tsai, W. Chou, Y. Su, Y. Lin, C. Sung, H. Dai, I. T. Yeh, W. Ku, T. Sung, W. Hsu, BIOSMILE: A semantic role labeling system for biomedical verbs using a maximum-entropy model with automatically generated template features, BMC Bioinformatics 8. doi:10.1186/

1471-2105-8-325.
URL https://doi.org/10.1186/1471-2105-8-325

[50] R. T. K. Lin, J. L. Chiu, H. Dai, R. T. Tsai, M. Day, W. Hsu, A supervised learning approach to biological question answering, Integrated Computer-Aided Engineering 16 (3) (2009) 271–281. doi:10.3233/ICA-2009-0316.
URL https://doi.org/10.3233/ICA-2009-0316

Appendix A Manual Evaluation on BioASQ 2017 Dataset

Question	Which enzyme is inhibited by niraparib?
Gold Answer	Poly(ADP-ribose) Polymerase
System's Answer	PARP
Comment	PARP is the abbreviation.
Question	How many cysteines have alpha-defensins?
Gold Answer	Alpha defensins contain six conserved cysteines
System's Answer	six
Comment	Question expects a number as answer.
Question	What is trichotillomania?
Gold Answer	Trichotillomania is a hair pulling disorder.
System's Answer	hair pulling disorder
Comment	The first part of the answer is redundant
Question	Viliuisk encephalomyelitis is diagnosed in which geographical area?
Gold Answer	Northeast Siberia
System's Answer	Siberia
Comment	Northeast is extra detail and it is not required.
Question	Which ApoE isoform is associated with hyperlipoproteinemia?
Gold Answer	ApoE2 isoform
System's Answer	ApoE2

Comment	Isoform is redundant word.
Question	Which is the largest metabolic gene cluster in yeast?
Gold Answer	The DAL cluster
System's Answer	DAL
Comment	cluster is redundant word.
Question	What fruit causes Jamaican vomiting sickness?
Gold Answer	Ackee fruit
System's Answer	Ackee
Comment	Fruit is redundant word
Question	What condition is usually represented by the acronym SUDEP?
Gold Answer	Sudden Unexpected Death in Epilepsy (SUDEP)
System's Answer	Sudden Unexpected Death in Epilepsy
Comment	SUDEP is not required.
Question	Which is the main cause of the Patau syndrome?
Gold Answer	Trisome 13
System's Answer	Trisomy 13
Comment	Trisomy is found in the passages.
Question	Which mutated gene causes the Ch0̆0e9diak2̆013Higashi Syndrome?
Gold Answer	LYST gene
System's Answer	LYST

Comment	Gene is redundant word.
Question	What is a miR?
Gold Answer	MiRs are small (23 nt) noncoding RNAs”
System’s Answer	MicroRNa
Comment	MicroRNA corresponds to MiR.
Question	What organism causes tularemia?
Gold Answer	Francisella tularensis
System’s Answer	F. tularensis
Comment	The system’s answer is a variation of the gold.
Question	Which disease is treated with lucinactant?
Gold Answer	respiratory distress syndrome
System’s Answer	RDS
Comment	RDS is the abbreviation.
Question	Which mushroom is poisonous, Amanita phalloides or Agaricus Bisporus
Gold Answer	Amanita phalloides
System’s Answer	Amanita
Comment	Phalloides is redundant word.
Question	Which is the most common gene signature in Rheumatoid Arthritis patients?
Gold Answer	IFN signature
System’s Answer	IFN

Comment	Signature is redundant word.

Table 9: Manual Evaluation on BioASQ 2017. Each row contains the question body along with the expected correct answer (gold answer). In addition, the system's output is presented along with a comment that justifies why the system's output is a variant of the gold answer.