Entity-Supported Summarization of Biomedical Abstracts

Frederik Schulze and Mariana Neves
Hasso-Plattner Institute
August-Bebel-Str. 88
Potsdam, 14482 Germany
mariana.neves@hpi.de

Abstract

The increasing amount of biomedical information that is available for researchers and clinicians makes it harder to quickly find the right information. Automatic summarization of multiple texts can provide summaries specific to the user's information needs. In this paper we look into the use named-entity recognition for graph-based summarization. We extend the LexRank algorithm with information about named entities and present EntityRank, a multi-document graph-based summarization algorithm that is solely based on named entities. We evaluate our system on a datasets of 1009 human written summaries provided by BioASQ and on 1974 gene summaries, fetched from the Entrez Gene database. The results show that the addition of named-entity information increases the performance of graph-based summarizers and that the EntityRank significantly outperforms the other methods with regard to the ROUGE measures.

1 Introduction

There is an overload of textual information, also in the biomedical domain, where new research articles are published daily. Text summarization can support to deal with this textual data deluge by providing automatically generated summaries on certain topics, e.g., a gene or a disease, as well as supporting answers returned by question answering (QA) systems.

However, the adoption of these technologies in the biomedical domain is not straightforward. The domain specific language has different requirements for information extraction compared to news articles, where *who, when, what, and where* elements are often the most important. Additionally, there are less resources available in biomedicine for text summarization, such as benchmarking corpora or knowledge bases. Finally, requirements for summaries, such completeness or correctness, are even more important in this domain when compared to others, as important decision might be taken based on them. Therefore, text summarization for biomedicine raises new challenges that still need to be addressed.

Searching for specific information in biomedical publications is a hard task that involves screening many entries in PubMed [1], the most popular search engine in biomedicine. PubMed contains over 24 million records and is growing exponentially (Lu, 2011). Two thirds of all queries to PubMed return more than 20 results, which is probably the reason why 47% of all queries get followed by a subsequent query without accessing any abstract or article of the search result (Dogan et al., 2009). In average, users read four documents to find the information they search for. Text summarization can support this task by providing summaries of many publications for a certain query or topic. Automatic text summarization has also the potential to support database curation by automatically generating short summaries about a topic, such as the ones manually created for the Entrez Gene database.

In this work, we propose two graph-based summarization algorithms based on named entities for improving automatic multi-document summarization for the biomedical domain. While the first approach

[1] https://www.ncbi.nlm.nih.gov/pubmed

Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM 2016), pages 40–49, Osaka, Japan, December 12th 2016.

extends the lexical PageRank (Erkan and Radev, 2004) with information from the NER, the second approach is solely based on named-entity recognition (NER). Additionally, we show how to adapt these algorithms for particular uses cases in biomedicine by including a bonus score. We evaluate our methods in two uses cases: generation of ideal answers for question answering (QA) systems and for gene summaries.

The remainder of this paper is structured as follows: next section presents related work on text sumarization. Section 3.2 introduces our summarization algorithms, followed by an evaluation of these in section 4. Finally, we present a discussion on the results, identify the limitations of our work, and propose future work in section 5.

2 Related work

Automatic text summarization has been studied since the late 1950s (Luhn, 1958) and has been applied to many domains such as news or social media, with some remarkable success (Chen et al., 2015). More recently, graph-based ranking algorithms like Google's PageRank (Page et al., 1999) and the HITS algorithm (Kleinberg et al., 1999) have been also successfully used for summarization. Essentially, these algorithms aim at deciding the importance of a vertex within a graph.

Mihalcea and Tarau introduced the TextRank (Mihalcea and Tarau, 2004) model, which essentially provides a general instruction for extracting lexical or semantical graphs from documents. In parallel and independent of the TextRank, Erkan and Radev introduced the LexicalPageRank (Erkan and Radev, 2004). One of the main changes that they proposed is the creation of multi-document summaries, by building a single sentence graph from multiple documents.

There are some previous work on summarization for biomedicine, such as (Kogilavani and Balasubramanie, 2009) that relied on ontologies to suport document retrieval and documents clustering. Researchers have also connected a graph-based extractive summarization algorithm with the domain knowledge of an ontology, for instance, for single document summarization (Morales et al., 2008). Their approach was similar to (Verma et al., 2007), but used a graph-based algorithm instead. Some researchers explored other non-traditional methods to generate automatic summaries. For example, generating not only summaries, but also a table of relevant data for extracting medical events and date times from documents (Aramaki et al., 2009). Another work proposed a system that uses patient data to provide a summarization of relevant information (Elhadad and McKeown, 2001).

Regarding summaries to support QA systems, some works focused on answering one special question type (e.g., "What is the best drug treatment for X?") (Demner-Fushman and Lin, 2006). Further, the general-purpose BioSquash system BIOSQUASH (Shi et al., 2007) was extended with biomedical domain knowledge from UMLS. The BioASQ challenge (Tsatsaronis et al., 2015; Krithara et al., 2016) was an important step to boost summarization solutions for biomedical QA systems, such as using machine learning approach based on Inductive Logic Programming (ILP) with different sets of features (Malakasiotis et al., 2015).

One of the first systems for automatic generation of gene summaries was proposed by (Ling et al., 2006). A different approach, that has some similarities to our work, was proposed by Jin et al. (Jin et al., 2009). They also use LexRank and extend it with two domain specific steps: identification of signature terms and calculation the similarity between each sentence based on the Gene Ontology (GO) terms. This approach is similar to our redundancy reduction step (cf. Section3.2). Finally, a similar approach to ours is the work of (Shang et al., 2014) which makes uses of TextRank based on frequency of words and LDA for topic relevance.

3 Methods and Materials

In this section, we introduce the data that we used and describe our methods for automatically generating summaries from scientific biomedical abstracts.

3.1 Data

Since we have two different use cases, we will also use two different datasets. However, both datasets rely on PubMed as source for the documents.

PubMed. PubMed [2] is a free search engine, that not only offers access to the MEDLINE database, but also to life science journals and online books. PubMed contains over 26 million records from over 26.000 journals.

Domain dictionaries. Our dictionaries combine data from the Unified Medical Language System[3] (UMLS), a collection of various health and biomedical vocabularies, and from SNOMED, a suite of standards from the U.S. Federal Government for the electronic exchange of health information. We specified a default name and a list of aliases and variations for each entity. Furthermore, entities were grouped by types, e.g., genes or diseases.

EntrezGene. EntrezGene[4] is a database for genes from various species. Each entry in EntrezGene is provided with a rich range of information[5], such as official symbol, corresponding organism and a short manually created summary.

3.2 Methods

In this section, we will describe the details of our summarization system and the algorithms that we have implemented.

Document retrieval. We fetched from PubMed the documents that we used for the generation of the summaries. We used the Entrez Programming Utilities [6], a set of public APIs that provide access to the data in PubMed. It allows users to get a full record by its PubMed identifier (PMID), as well as to retrieve documents which match some given keywords. Overall, we fetched 68.083 abstracts, which were used to generate summaries. We rely only on the abstracts, not on the full text, due to the following reasons: (a) most of the records in PubMed contain only an abstract; (b) sentences in an abstract are more suitable for a summary, since they are a summarization of an article and contains the most relevant information; (c) the BioASQ dataset (cf. section 4.1) are based on the abstracts, not on the full text.

Document pre-processing. We extracted linguistic and semantic annotations for the documents using the built-in text analysis functionality of an in-memory database (SAP HANA). Therefore, we created two full-text indexes (FTI), i.e., a a full indexing of all documents, one for linguistic annotations, i.e., part-of-speech (POS) tags and stems, and one for semantic annotations, i.e., named entities for genes, diseases, etc. The name-entity recognition (NER) was based on the custom dictionaries derived from various terminologies from UMLS, as previously described in our question answering system (Schulze et al., 2016). Dictionary matching to the documents was performed inside the database based on an approximated matching of the dictionaries with the documents. We mapped the words in the linguistic index to the entities in the NER index only for entities composed of a single word.

Extended LexRank. We extended the LexRank graph-based algorithm (Erkan and Radev, 2004) with information from the NER step. LexRank finds the most central sentences by building a sentence graph, based on the idf-modified cosine similarity, and calculates the PageRank on the resulting graph. We relied on the pre-processing step to normalize each sentence, based on the stemming and the named entities. After running the LexRank on these normalized sentences, we remove redundant sentences and extract the best sentences for a summary.

LexRank is based on the assumption that similar sentences contain the exact same words. However, two sentences can express the same content using different forms of the same word or even their syn-

[2] http://www.ncbi.nlm.nih.gov/pubmed
[3] http://www.nlm.nih.gov/research/umls/
[4] http://www.ncbi.nlm.nih.gov/gene
[5] e.g., humangeneHNF1A, http://www.ncbi.nlm.nih.gov/gene/6927
[6] http://www.ncbi.nlm.nih.gov/books/NBK25501/

onyms. We rely on stems, from the linguistic FTI, and on the named entities from the semantic FTI to normalize different but related words and to reduce the dimension of the vector space.

After merging both indices, we unify the resulting index by choosing a single form from one of the available ones. For instance, only 61% of the words were recognized as named entities. Therefore, we choose one form in this order: named entity, stem, normalized form or token text. Since the named entities not only include different forms of the same word but also synonyms and typical spelling mistakes, we chose this as the preferred form, followed by the stemmed form, that could also include different word classes and conjugations. The so-called normalized form only accounts for capitalization and mutated vowels, while the token text is the original word as it appears in the text. The unified index was used as input for the LexRank algorithm and for the calculation of the idf values.

After using LexRank to rank the sentences according to their centrality, we need additional post-processing steps to generate the summaries. Firstly, we extracted the sentences that are most suitable for a summary using the following process: (1) we initialized two sets: an empty set A and a set B that contains all extracted sentences; (2) we ordered the sentences in set B by descending order according to their score; (3) we moved the top sentence s_i from set B to set A. Then penalized all sentences s_j similar to s_i by calculating their new score according to the equation 1 below, where $sim(s_i, s_j)$ is the similarity between two sentences, $t = 0.3$ is a threshold and $w = 0.5$ is a penalty factor; (4) we repeated steps 2 and 3 until enough sentences were in set A.

$$score(s_j) = \begin{cases} w \times score(s_j), & \text{if } sim(s_i, s_j) \geq t \\ score(s_j), & \text{otherwise} \end{cases} \tag{1}$$

At the end of this procedure, we obtained the most central sentences that are also as distinct as possible to each other. We concatenated these sentences to create a summary with the most relevant information.

EntityRank. We developed a second ranking approach inspired by the LexRank. Similar to LexRank, EntityRank also uses the similarity between the sentences to generate a sentence graph, but based on the named entities. Additionally, it also includes the possibility to adapt the calculation to the specific use cases.

Since the EntityRank is a graph-based algorithm, like LexRank and PageRank, we built a graph from the documents that we want to summarize. In order to represent the text as a graph, we created a similarity matrix by comparing every two sentences to each other, with no distinction between sentences that came from the same or from different documents. We experimented with two graph approaches, namely, weighted and non-weighted edges, and we implemented three approaches.

Our first approach used a non-weighted graph, like in LexRank, by adding non-weighted edges between the vertices that have a similarity greater than a certain threshold. The value of this threshold directly influences the density of the resulting graph, since a lower threshold results in more connections. After evaluating various values, we decided for 0.2. For the second approach, we created a weighted sentence graph. Compared to the unweighted graph, this method has no loss of information, since we add an edge between every two vertices whose corresponding sentences have a similarity greater than zero. This usually resulted in a much larger and dense graph, but it also contained much more information. We calculated the similarity between two sentences based on the cosine similarity and on the named entities. Our third hybrid approach combined the threshold used in the first method with the use of weighted edges from the second method. We add a weighted edge between every two sentences whose similarity score is larger than a certain threshold. We decided for a value of 0.1 for the threshold based on our experiments.

After creating the sentence graph, we calculated a score that represents the centrality of the sentence based on PageRank (Page et al., 1999), which is a round-based algorithm. We recalculated the score of every vertex in each round by using the results from the previous round until convergence of the scores, using the equation below:

$$\text{score}(s_i) = \frac{d}{N} + (1-d) \sum_{s_j \in adj[s_i]} \frac{\text{score}(s_j)}{deg(s_j)} \tag{2}$$

where N is the total number of vertices in the graph, $adj[s]$ are all adjacent vertices of the vertex s, $deg(s_j)$ is the degree of vertex s_j, and d is a 'damping factor', which is typically set between 0.1 and 0.2 as proposed by (Page et al., 1999).

Since this formula is only defined on non-weighted graphs, we did not use it on weighted graphs. Instead, we modified it to distribute the score of each vertex, depending on the weights of its edges. The resulting equation is a modified version of the EntityRank for weighted graphs:

$$\text{score}(s_i) = \frac{d}{N} + (1-d) \sum_{s_j \in adj[s_i]} \frac{cosine(s_i, s_j)}{\sum_{s_y \in adj[s_j]} cosine(s_y, s_j)} score(s_i) \tag{3}$$

where $cosine(s_1, s_2)$ is the cosine similarity between two sentences. With this formula, we calculated the non-weighted EntityRank and the weighted EntityRank using the same weight for every edge.

Similar to the PageRank, that gives more importance for certain Web sites, we introduced a score to describe the relevance of a sentence for a use case. This score was used to change the distribution of EntityRank and to give an higher weight to the sentences that have a greater relevance for the use case. We changed the equation accordingly:

$$\text{score}(s_i) = E(s_i)\frac{d}{N} + (1-d) \sum_{s_j \in adj[s_i]} \frac{cosine(s_i, s_j)}{\sum_{s_y \in adj[s_j]} cosine(s_y, s_j)} score(s_i) \tag{4}$$

where $E(s_i)$ is the relevance of the sentence s_i for the current use case. The average of this factor over all vertices is 1, in order to keep the average of all scores converging to 1.

EntityRank for question answering. Given that summaries for QA systems should focus on the question, we changed EntityRank to provide a bonus score for sentences that are related to the question. The bonus score was calculated based on the similarity of each sentence with the question, more specifically, on the common named entities. But before we can use the similarity, we need to normalize it, so that its average over all sentences is 1, by using the following equation:

$$E(s_i) = sim(s_i)\frac{|S|}{\sum_{j \in S} sim(s_j)} \tag{5}$$

where S is the set of sentences and $sim(s_i)$ the similarity of the sentence s_i to the question. We will use this normalized bonus score in the equation 4 to positively influence the summarization.

EntityRank for gene summaries. Since there is lot of information about most of the genes, the algorithm needs additional guidance on the right information for the summary. When analyzing human summaries from the Entrez Gene database, we noticed that they all cover similar topics, such as encoded protein, mutations, location or relation with certain diseases.

We created a bonus score that reflects how suited a sentence is for belonging to a gene summary. We relied on 9553 manually written summaries from EntrezGene to identify the most frequent named entities. The 15 most used terms were the following: "Proteins", "Genes", "protein location", "encoding", "variant", "family", "last name", "variant", "receptor", "receptor cells", "mutation", "numerous", "function", "enzymes", and "DNA". We created an artificial sentence from these most used terms, then we compared the artificial sentences to the sentences in the dataset. This similarity score was used to create the bonus score using equation 5 that was later applied in equation 4.

4 Evaluation

Unlike other domains, there are not gold-standard dataset for the evaluation of text summarization for biomedicine. Most papers introduce their own evaluation datasets, based either on abstracts from documents or on manual evaluation by experts. In this section, we evaluate our algorithms on the test collection provided by BioASQ and on a set of human summaries from the EntrezGene database. Tuning of parameters in our methods were solely based on the training sets.

4.1 Datatsets

BioASQ We used 1009 questions and the corresponding data provided by the BioASQ challenge [7], an EU-funded project that aims to evaluate biomedical QA. These questions correspond to the datasets used in phase B of the tasks 1b (2013), 2b (2014) and 3b (2015). The BiOASQ dataset includes not only the questions, but also the relevant documents (PMIDs from PubMed), passages and concepts, as well as exact answer and ideal answers (summaries). We relied on the these relevant documents (but not the passages) to generate our summaries. These datasets may include more than one ideal answer for each question, which we use for the evaluation of our system.

EntrezGene summaries We automatically collected manual summaries from EntrezGene. The resulting summaries include some noise, therefore, we removed all summaries that were shorter than 100 characters and those which were equal for multiple genes. We then split this set randomly into a training set of 9553 summaries, that was used for generating the bonus score, and a test set of 1974 summaries. The quality of this dataset is very low, when compared to the BioASQ data. Further, the summaries have different lengths and some might be outdated. In the document retrieval step, we relied on the PubMed API to search for the official symbol, official full name and the whole name of the gene. We considered the top 20 abstracts for each gene as source for the summary.

4.2 Results

We compared our summarization algorithms to LexRank. Further, we also evaluated the performance of EntityRank for our two use cases. We used the Java implementation of ROUGE-N from the Dragon Toolkit (Zhou et al., 2003), which was developed by (Zhou et al., 2007). It implements ROUGE-1 and ROUGE-2 with additional stop word removal. Since ROUGE is a recall-based measure, the length has no influence on the score. Therefore, we calculated the ROUGE scores on summaries of similar length.

Comparison to LexRank. We use the BioASQ dataset for comparison to LexRank, given its better quality. As recommended by BioASQ, we created summaries with a fixed length of 200 words and compared them to the reference summaries using both ROUGE-2 and ROUGE-1. Additionally, we also compared the algorithms for shorter summaries of only 100 words. Therefore, we extracted sentences until the next sentence would not fit in the limit of words.

Our comparison to LexRank is displayed in Figure 1. We compared LexRank to the extended LexRank, the non-weighted and the weighted EntityRank, without considering any bonus scores. In these diagrams, the blue bar shows the average score of all summaries, which adds up to 1009 summaries, while the green bar shows the average score only for summaries which were created from more than 20 documents, which adds up to 197 summaries. Results for ROUGE-2 (results not shown) had a similar correlation among the various systems, though lower results.

Our extended LexRank achieved a slightly better score, compared to the original LexRank. This was expected, since the algorithm is essentially the same. In contrast, the non-weighted and weighted variants of the EntityRank produced very different results. The overall score of the non-weighted EntityRank is far lower, compared to the weighted EntityRank. This can be explained by the loss of information that occurs when ignoring the similarity between the sentences. Finally, we got better ROUGE scores for the summaries that were created with the weighted EntityRank.

[7] www.bioasq.org

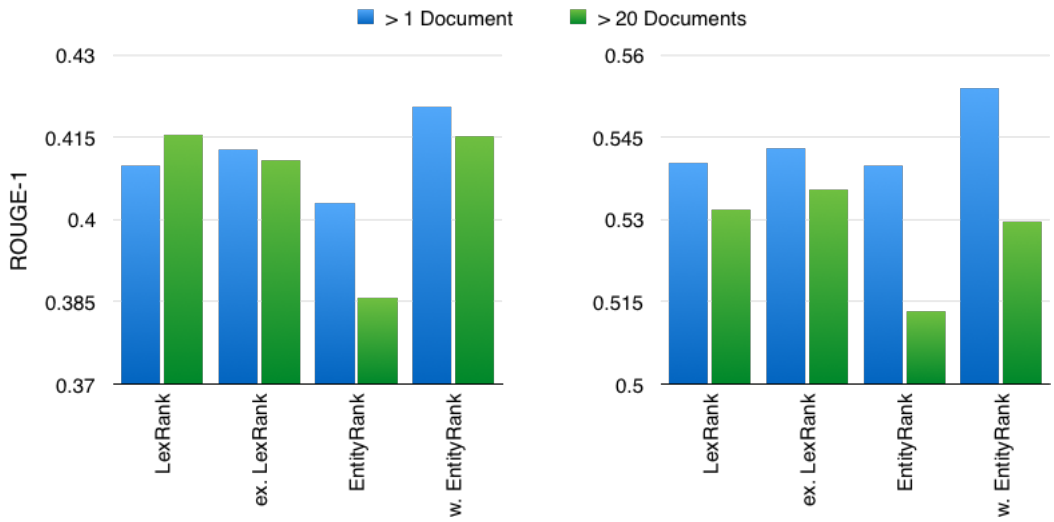

Figure 1: ROUGE-1 scores for 100-words summaries (left) and 200-words summaries (right).

Question Answering. We evaluated the influence of the bonus factor by comparing the adapted version of the weighted EntityRank with the general version. Therefore we generated 100- and 200-words summaries and compared them according to the number of abstracts that were used for the generation (cf. Figure 2). The bonus score slightly improved the results of the EntityRank, especially for summaries that were generated from fewer abstracts. While the 100-words summaries benefited from the bonus score, regarding summaries that were generated from less than 10 abstracts, 200-words summaries were improved or were similar for each number of abstracts. Although the overall improvement is only 2%, it shows that the bonus score did help guiding the EntityRank on the right sentences.

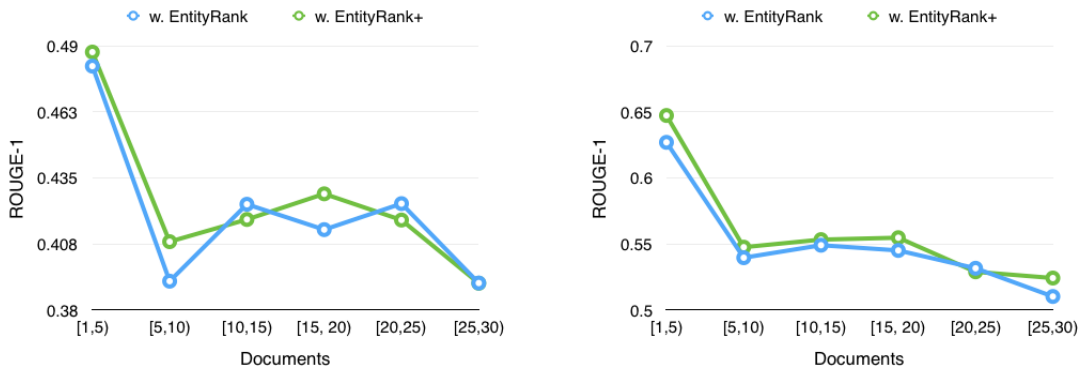

Figure 2: Comparison of ROUGE-1 scores for 100-words summaries (left) and 200-words summaries (right) of the weighted EntityRank (with and without the bonus score).

We also compared our systems to participants of the BioASQ challenge during the current fourth edition of the challenge (Schulze et al., 2016). We participated with the basic EntityRank and generated summaries with a length of five sentences instead of 200 words, which was the best version of our system that was ready at the time of the challenge (Spring/2016). As reported in (Schulze et al., 2016), we got a first position in one of the batches and good scores in the other batches.

Gene Summarization. We also evaluated whether our bonus score could improve the generation of gene summaries. We evaluated the same algorithms used earlier in the comparison to LexRank, as well as weighted and unweighted EntityRank with the bonus score (cf. Figure 3). The overall scores are significantly lower compared to the ROUGE-1 scores of the BioASQ dataset. This is due to the fact that the gene summaries is not a query-focused task and it can include many different topics related to the gene, while the summaries for QA should be related to the query (question). The weighted EntityRank

obtained the highest scores among all systems. Thus, we can confirm the positive influence of the bonus score, which increased the scores of unweighted, as well as the weighted, EntityRank by roughly 2%.

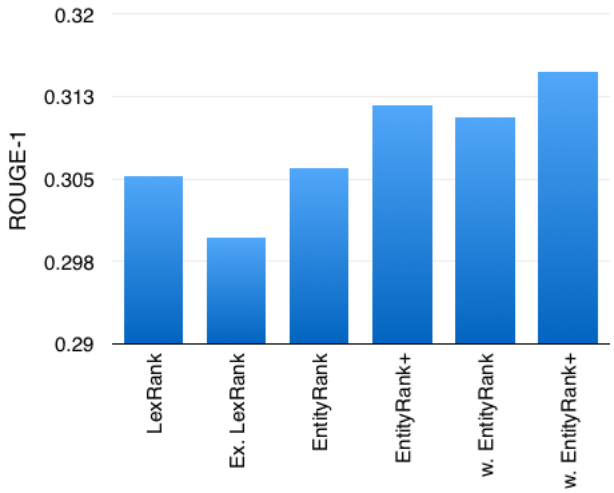

Figure 3: Comparison of ROUGE-1 scores for 200-words gene summaries.

5 Discussion and Future Work

Considering that summaries need dense sentences with much information, we chose to use only the abstracts of the PubMed articles. But especially for a focused summary, there could be information in the full paper that is not contained in the abstract. Therefore, in future work, we plan to evaluate the performance of our algorithms on full text, specially regarding the gene summaries, as information contained on these are not restricted to the abstracts.

The ordering of the sentences and the readability are important issue to create a fluent and natural summaries similar to human-written ones (Jurafsky and Martin, 2009). Further, it has a large impact on the comprehensiveness of a summary. For example, the information of a sentence could require knowledge of another sentence.

Further, we did not evaluate the performance of our NER approach for the detection of biomedical terms. The reason for not recognizing a term could not only be a failure of the NER step but could also be due to misspelling of words or missing terms in our dictionaries. Especially the latter could possibly occur more often during real use, if new documents contain words that is still not available in our dictionaries. Since our algorithms are heavily dependent on the named-entities, we need to check whether there is still room for improvement on the NER step, either regarding adding new terminologies or using machine learning approaches.

Not only false negatives are an issue, but also false positives. False positives can be either due to the NER algorithm or an error in the dictionaries. This is a problem specially for acronyms which often match common English words, such as conjunctions, when using an approximate and lowercase-based matching. A stopwords filtering step could remove some of these false positives.

Finally, one issue could raise from merging the linguistic and semantic indices. Set of words that have the same stem and a related meaning could have been handled as the same. But this was not the case if one of them was recognized as a named-entity. However, we anticipate that these were indeed rare cases.

6 Conclusions

Automatic text summarization has the potential to support many domains. It enable researchers to quickly get an overview of a specific topic, without investing too much time for searching the information. Especially for fast changing domains like biomedicine, it can help clinicians to save valuable time that could be use on treating patients. Therefore, summarization algorithms should utilize domain knowledge to create accurate summaries.

In order to improve the understanding of the domain, we relied on named-entity recognition in our summarization algorithms. We showed that adding named entities to graph-based summarization algorithm did improve the results for the task. The named-entity information supported building a sentence graph, by improving the similarity measure. This approach is especially effective for summaries that are generated from few documents.

We showed two different ways of enriching graph-based algorithms with named-entity information. The extended LexRank proved that using named entities greatly improves the overall results of the LexRank in the medical domain. But the main contribution of this work is EntityRank, which is a graph-based algorithm that is solely based on named entities instead of terms. We showed that a graph-based summarization algorithm that only uses technical terms can outperform other graph-based approaches within a restricted domain. Although the performance only increases for summaries that are created from few documents, it still shows the potential of that approach, since it performs comparable to other state-of-the-art systems.

Additionally, we showed how the EntityRank could be adapted to more specialized use cases, such as question answering and gene summarization. We implemented two bonus scores and showed how they improve the results for special use cases.

References

Eiji Aramaki, Yasuhide Miura, Masatsugu Tonoike, Tomoko Ohkuma, Hiroshi Mashuichi, and Kazuhiko Ohe. 2009. Text2table: Medical text summarization system based on named entity recognition and modality identification. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 185–192. Association for Computational Linguistics.

Kuan-Yu Chen, Shih-Hung Liu, Berlin Chen, Hsin-Min Wang, Ea-Ee Jan, Wen-Lian Hsu, and Hsin-Hsi Chen. 2015. Extractive broadcast news summarization leveraging recurrent neural network language modeling techniques. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 23(8):1322–1334.

Dina Demner-Fushman and Jimmy Lin. 2006. Answer extraction, semantic clustering, and extractive summarization for clinical question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 841–848. Association for Computational Linguistics.

Rezarta Islamaj Dogan, G Craig Murray, Aurélie Névéol, and Zhiyong Lu. 2009. Understanding pubmed® user search behavior through log analysis. *Database*, 2009:bap018.

Noemie Elhadad and Kathleen R McKeown. 2001. Towards generating patient specific summaries of medical articles. In *In Proceedings of NAACL-2001 Workshop Automatic*. Citeseer.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, pages 457–479.

Feng Jin, Minlie Huang, Zhiyong Lu, and Xiaoyan Zhu. 2009. Towards automatic generation of gene summary. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 97–105. Association for Computational Linguistics.

Dan Jurafsky and James H. Martin. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson international edition. Prentice Hall.

Jon M Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S Tomkins. 1999. The web as a graph: measurements, models, and methods. In *Computing and combinatorics*, pages 1–17. Springer.

AA Kogilavani and B Dr P Balasubramanie. 2009. Ontology enhanced clustering based summarization of medical documents. *International Journal of Recent Trends in Engineering*, 1(1).

Anastasia Krithara, Anastasios Nentidis, Georgios Paliouras, and Ioannis Kakadiaris. 2016. Results of the 4th edition of bioasq challenge. In *Proceedings of the Fourth BioASQ workshop at the Conference of the Association for Computational Linguistics*, pages 1–7.

Xu Ling, Jing Jiang, Xin He, Qiaozhu Mei, Chengxiang Zhai, and Bruce Schatz. 2006. Automatically generating gene summaries from biomedical literature.

Zhiyong Lu. 2011. Pubmed and beyond: a survey of web tools for searching biomedical literature. *Database*, 2011:baq036.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

Prodromos Malakasiotis, Emmanouil Archontakis, Ion Androutsopoulos, Dimitrios Galanis, and Harris Papageorgiou. 2015. Biomedical question-focused multi-document summarization: Ilsp and aueb at bioasq3. In *Working Notes for the Conference and Labs of the Evaluation Forum (CLEF), Toulouse, France*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. Association for Computational Linguistics.

Laura Plaza Morales, Alberto Díaz Esteban, and Pablo Gervás. 2008. Concept-graph based biomedical automatic summarization using ontologies. In *Proceedings of the 3rd textgraphs workshop on graph-based algorithms for natural language processing*, pages 53–56. Association for Computational Linguistics.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: bringing order to the web.

Frederik Schulze, Ricarda Schler, Tim Draeger, Daniel Dummer, Alexander Ernst, Pedro Flemming, Cindy Perscheid, and Mariana Neves. 2016. Hpi question answering system in bioasq 2016. In *Proceedings of the Fourth BioASQ workshop at the Conference of the Association for Computational Linguistics*, pages 38–44.

Yue Shang, Huihui Hao, Jiajin Wu, and Hongfei Lin. 2014. Learning to rank-based gene summary extraction. *BMC bioinformatics*, 15(Suppl 12):S10.

Zhongmin Shi, Gabor Melli, Yang Wang, Yudong Liu, Baohua Gu, Mehdi M Kashani, Anoop Sarkar, and Fred Popowich. 2007. Question answering summarization of multiple biomedical documents. In *Advances in Artificial Intelligence*, pages 284–295. Springer.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):138.

Rakesh Verma, Ping Chen, and Wei Lu. 2007. A semantic free-text summarization system using ontology knowledge. In *Proc. of Document Understanding Conference*.

Xiaohua Zhou, Xiaodan Zhang, and Xiaohua Hu. 2003. The Dragon Toolkit. `dragon.ischool.drexel.edu` [Online; accessed 14-June-2016].

Xiaohua Zhou, Xiaodan Zhang, and Xiaohua Hu. 2007. Dragon toolkit: incorporating auto-learned semantic knowledge into large-scale text retrieval and mining. In *Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on*, volume 2, pages 197–201. IEEE.