Extracting highlights of scientific articles: A supervised summarization approach

Extracting Highlights of Scientific Articles: a Supervised Summarization Approach

Luca Cagliero*, Moreno La Quatra

Dipartimento di Automatica e Informatica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129, Torino, Italy

Abstract

Scientific articles can be annotated with short sentences, called highlights, providing readers with an at-a-glance overview of the main findings. Highlights are usually manually specified by the authors. This paper presents a supervised approach, based on regression techniques, with the twofold aim at automatically extracting highlights of past articles with missing annotations and simplifying the process of manually annotating new articles. To this end, regression models are trained on a variety of features extracted from previously annotated articles. The proposed approach extends existing extractive approaches by predicting a similarity score, based on n-gram co-occurrences, between article sentences and highlights. The experimental results, achieved on a benchmark collection of articles ranging over heterogeneous topics, show that the proposed regression models perform better than existing methods, both supervised and not.

Keywords:
Highlight extraction, Extractive summarization, Regression models, Text mining and analytics

*Corresponding author. Tel.: +39 011 090 7179. Fax: +39 011 090 7099.
Email addresses: luca.cagliero@polito.it (Luca Cagliero), moreno.laquatra@polito.it (Moreno La Quatra)

1. Introduction

Extractive document summarization entails automatically identifying the most significant content from the body of a document (Nenkova & McKeown, 2012). Unlike abstractive approaches, whose main goal is to generate new content related to the existing one, extractive methods focus on picking keywords, keyphrases, or entire sentences from the original documents. Summarization algorithms have found application in a number of contexts among which news summarization (Giannakopoulos, 2013) and tweet summarization (Naik & Bojewar, 2017). This paper studies the application of extractive summarization techniques to scientific articles.

Generating summaries of scientific articles entails addressing any of the following problems: (i) identify the keywords that describe the main topics covered by an article, (ii) generate an abstract of an article, or (iii) select the content that is most likely to appear in the article highlights. Keywords are single words or phrases (i.e., combinations of salient words), which are usually extracted using keyphrase extraction techniques (Hasan & Ng, 2014). Abstracts are concise descriptions of the most salient document content and consist of a sequence of full sentences. Although they are commonly available for most of the published articles, they can be also generated using extractive summarization methods (Nikolov et al., 2018). Highlights are short sentences, which are typically manually annotated by the article authors. They provide readers with an at-a-glance, result-oriented overview of the main findings of the article. Since their purpose is to provide a quick snippet of the results achieved in the presented research work, highlight content may differ from that of the article abstract.

This work addresses task (iii) by using a sentence-based approach, i.e., it identifies a subset of article sentences whose content is worth including in the article highlights. Notice that unlike the abstract, which is usually part of the article itself, highlights are not required to exactly match any sentence in the paper. However, we can assume, to a good approximation, that there exist sentences in the analyzed article that cover, to a large extent, all the aspects mentioned by the authors in the highlights.

For example, Table 1 contains the abstract of a representative paper, the highlights given by the authors (consisting of three manually written sentences), and the selections of article sentences produced by different summarization approaches (whose characteristics will be discussed later on). The scope of the highlights appears to be significantly different from those of the

abstract. Highlights are separate sentences that summarize the main contribution of the article and its main achievements, whereas the abstract is an organized sequence of sentences, where all the aspects covered by the article are mentioned. Therefore, summarization systems that produce high-quality abstracts are not necessarily suitable for extracting highlight content.

Recalling the examples reported in this paper in Table 1, the summary produced by the established graph-based method contains less concise and focused sentences than those produced by regression approach (denoted as *best algorithm*).

Since the process of annotating articles with highlights has recently been introduced, for many articles the corresponding highlights are still missing.

The aim of the approach presented in this paper is twofold: (i) Support the authors of new scientific articles in manually annotating articles with meaningful highlights. It recommends a subset of article sentences containing relevant content. (ii) Automatically generate the highlights of past articles with missing annotations. The system proposed in this work relies on regression models trained on an heterogeneous collection of previously annotated articles. To capture the most relevant sentence-level information and its correlation with the content of the highlights, we extract and include in the training dataset a variety of features describing the relevance of the terms occurring in the sentence, the position of the sentence in the article, the similarity between sentences and abstract, and the significance of the sentences in latent spaces of sentence vectors. The regression model predicts the similarity between the sentences in the paper to be annotated and the highlights. The higher the similarity, the most likely the sentence contains information that is worth considering during the annotation process.

The performance of the proposed approach was compared with that of established summarization approaches, both supervised and not, on a recently released benchmark collection of scientific articles (Collins et al., 2017). The collection includes a variety of scientific articles ranging over different topics in the computer science domain. The results show that the proposed regression-based approach is superior to the existing approaches in order to tackle the highlight extraction problem.

In a nutshell, the innovative contribution of this paper can be summarized as follows:

- The paper proposes a regression-based approach to extracting highlights from scientific papers. To the best of our knowledge, this work

is the first attempt to use regressors instead of classification models.

- It trains the models on a variety of features describing the similarity between the candidate sentences and the expected highlights under different viewpoints.
- It compares the performance of the proposed strategy with that of many supervised and unsupervised approaches proposed in literature on a benchmark, general-purpose paper collection annotated with highlight information.
- It explores the performance of the systems on two additional datasets (which were made available to the research community for the sake of reproducibility), which consist of subject-specific papers.

The rest of the paper is organized as follows. Section 2 compares this work with the existing literature. Section 3 thoroughly describes the proposed method. Section 4 reports the main experimental results, while Section 5 draws conclusions and discusses future works.

2. Literature review

The previous works related to scientific article summarization addressed the following issues: (i) Abstractive generation of title/abstract information (Nikolov et al., 2018; Kim et al., 2016; Lloret et al., 2013). (ii) Extractive keyphrase extraction using classification techniques (Gollapalli & Caragea, 2014; Krapivin et al., 2010). (iii) Highlight extraction based on binary classification (Collins et al., 2017).

Abstractive generation of title/abstract of scientific articles. The approaches proposed by Kim et al. (2016); Lloret et al. (2013) first selected relevant words from the body of the article and then applied information fusion. For example, Lloret et al. (2013) applied this methodology in order to generate the abstracts of biomedical papers. Kim et al. (2016) explored the use of supervised techniques to generate intermediate results for abstractive summarization, i.e., they generated sentence-level summaries of each paragraph. The intermediate results are exploited to automatically generate title and abstract using different types of Neural Network models. Unlike Nikolov et al. (2018); Kim et al. (2016); Lloret et al. (2013), this work focuses on extractive summarization rather than on an abstractive approach.

Keyphrase extraction from scientific articles. Krapivin et al. (2010) extracted keyphrases from scientific documents by combining classification and Natural Language Processing (NLP) techniques. Specifically, they first extracted relevant text features (e.g., POS tags). Then, they trained classification models on the extracted features. Their results show that integrating NLP features allows them to significantly improve the performance of classification techniques. To extend the set of features related to scientific articles, the work presented by Gollapalli & Caragea (2014) proposed to integrate citation network information. The aim of the approaches presented by Gollapalli & Caragea (2014); Krapivin et al. (2010) is significantly different from those addressed in this work, which specifically focuses on extracting highlights by means of a sentence-based summarization approach.

Highlight extraction from scientific articles. An attempt to extract highlights from scientific articles has already been made by Collins et al. (2017). The authors identified the most relevant sentences from the article full-text by applying a binary classifier. Furthermore, they presented a new benchmark dataset, which consists of 10,000 articles annotated with the corresponding highlights. Although the approach presented by Collins et al. (2017) classifies article sentences as relevant or not for highlight generation, the proposed system does not produce a sentence ranking, i.e., a selection of the top-K most relevant sentences. This prompt the need for a new, supervised method which is able to identify and rank the candidate sentences. This paper aims at overcoming the above issue by applying regression models. To the best of our knowledge, this is the first attempt to apply multivariate regression methods in order to recommend a ranked list of sentences useful for annotating a scientific article.

Extractive summarization of other document types. A relevant effort has been devoted to extracting summaries, consisting of a subset of sentences, from other document types (e.g., news articles). Many previous works addressed the problem using unsupervised techniques, among which clustering algorithms (e.g., COSUM (Alguliyev et al., 2019)), graph-based techniques (e.g., CoreRank (Tixier et al., 2017), LexRank (Erkan & Radev, 2004), TextRank (Mihalcea & Tarau, 2004), Co-Rank (Fang et al., 2017)), itemset mining algorithms (ELSA (Cagliero et al., 2019), MWI-SUM (Baralis et al., 2015)), Latent Semantic Analysis (e.g., LSARank (Steinberger & Jezek, 2004), JRC (Steinberger et al., 2011), UWB (Steinberger, 2013)), and optimization-based approaches, e.g., decoding and integer linear programming algorithms based on the concepts of maximal coverage (Takamura

& Okumura, 2009; Gillick & Favre, 2009) and the Minimum Description Length (Litvak et al., 2015)). However, since unsupervised models are not tailored to the target under analysis, they potentially miss the correlation between the selected content and that of the expected highlights.

Parallel efforts have also been devoted to applying supervised models in order to extract news article summaries. For instance, Neural Network-based approaches have been presented by Nallapati et al. (2017) and Zhou et al. (2018)). A supervised approach based on genetic algorithms has been presented by Litvak et al. (2010), while Wong et al. (2008) integrated supervised and semi-supervised learning to train an extractive summarization algorithm able to retrieve relevant sentences according to extrinsic and content-related features. Recently, Mao et al. (2019) presented a single-document news summarization algorithm combining supervised and unsupervised techniques. They combined statistical and graph-based features to improve the performance of state-of-the-art methods. In this way, this paper investigates the use of supervised models to extract article highlights, which is a complementary, challenging, yet relevant research issue.

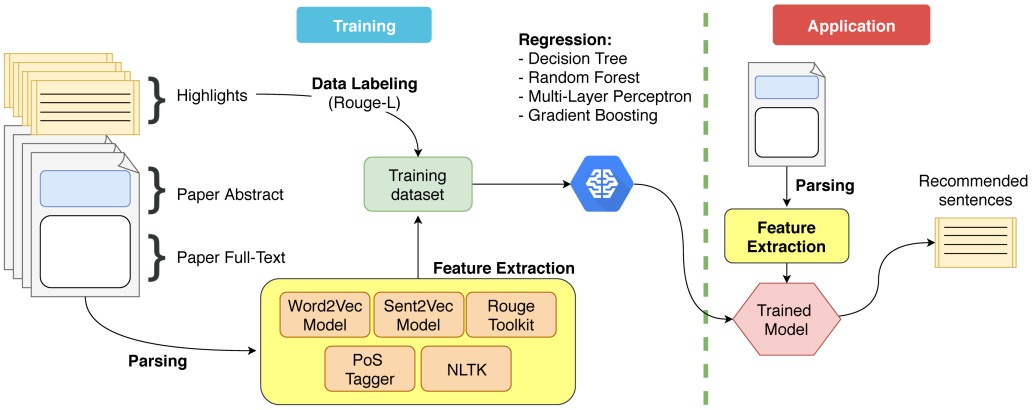
3. Proposed method

The method presented in this study identifies the top K sentences of a scientific article (where K is an analyst-provided parameter) whose content is most likely to be correlated with the article highlights. The proposed method has the twofold aim at supporting the manual annotation of new articles (by providing pertinent suggestions to the annotators) and automatically annotating past articles with missing highlight information. Sentence recommendation is based on predictive models trained on a collection of manually annotated articles.

The main steps of the data analytics process, depicted in Figure 1, are summarized below.

- *Feature extraction*: it extracts salient information from the full-text of a large set of annotated articles.
- *Sentence labeling*: it measures and stores the similarity between article sentences and highlights.
- *Model training*: it generates a regression model on the prepared dataset that describes the most significant correlations between the analyzed data features and the similarity score.

Figure 1: Sketch of the proposed methodology.



- *Model application*: it applies the model on test data in order to predict sentence-level scores and to rank sentences by decreasing similarity score.

A more thorough description of each data analytics step is given below.

3.1. Feature extraction

We analyze a collection of scientific articles $\mathcal{A}$. Each article in $\mathcal{A}$ has been manually annotated by domain experts with textual highlights. To characterize article content at the sentence level, we split the full-text of each article (excluding the abstract) into sentences using punctuation and we model each article $A_i \in \mathcal{A}$ as a set of distinct sentences $\{s_1^i, s_2^i, \ldots, s_n^i\}$. Similarly, the textual highlights associated with A_i are modelled as a set of sentences $\{h_1^i, h_2^i, \ldots, h_Q^i\}$. The number q of highlights per article typically ranges between 3 and 5 and is commonly specified by the editor.

The goal of the feature extraction process is to extract sentence-level information describing

- the most relevant textual properties related to syntax, semantics, and structure of the text,
- the frequency-based relevance of the words occurring in the sentence,
- the significance of sentence content in latent spaces,
- the similarity between the sentence and the abstract of the article.

All the feature categories mentioned above are deemed as potentially useful for predicting the similarity between article sentences and highlights. For each sentence in the analyzed articles, we collect the values of all the features into a relation dataset $\mathcal{R}$. Each record $r_i \in \mathcal{R}$ corresponds to a specific article and sentence (Rajaraman & Ullman, 2011). Hereafter, we will thoroughly describe the considered features and discuss the reasons why they are deemed as relevant in this particular context of analysis.

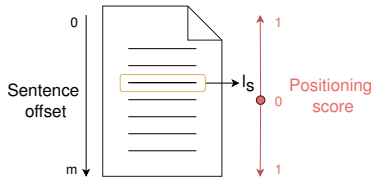
Symbols count. This feature indicates whether a sentence contains mathematical symbols or special characters. Since highlights rarely include formulas or technical details, sentences containing symbols are less likely to share significant content with the humanly generated highlights.

Parts Of Speech. These features indicate the presence in the sentence of singular and plural nouns, adjectives, conjunctions, proper nouns and present-form verbs. As pointed out by Krapivin et al. (2010), the predominance of specific POS types in the sentence is highly-correlated with its relevance. To compute feature values, we applied the POS-tagger provided by the Natural Language ToolKit (Bird et al., 2009).

Sentence position. This feature indicates the relative position of the sentence in the article. Most of the scientific articles have a common structure, i.e., they contain an abstract, an introductory section, a methodology description, a summary of the main results, and a separate section drawing the conclusions of the work. The distribution of the highlight information across the article sections is quite recurrent and could be deemed as relevant for identifying relevant sentences. Notice that unlike the content of the abstract, which is likely to be referred to mainly in the introductory and conclusive sections, highlight information is result-oriented thus it is likely to be mentioned throughout the sections of the article. Hence, the features describing the content position need to be tailored to the specific problem under analysis.

We consider as additional sentence-level feature (i) a categorical representation of the section to which the sentence belongs to and (ii) a positioning score $f(s)$ (ranging between 0 and 1), which is the position offset of a sentence s with respect to the middle of the section. Since technical details are more likely to be reported in the central part of a text, informative content is more likely to appear at the beginning or at the end of the sections (Krapivin et al., 2010). The score $f(s)$ of an arbitrary sentence s is computed according

to the formula reported below (where l_s is the sentence offset and m is the maximum offset in the considered article).



$$f(s) = \begin{cases} -\dfrac{l_s}{m/2} + 1 & \text{if } l_s \geq \frac{m}{2} \\ \\ +\dfrac{l_s}{m/2} - 1 & \text{if } l_s < \frac{m}{2} \end{cases}$$

Frequency-based word relevance. The sentences that include many highly relevant terms are most likely to contain highlight content. Term relevance can be evaluated using frequency-based statistics. Specifically, in this work we use the Term Frequency-Inverse Sentence Frequency (TF-ISF) statistics, which is among the most established and used term relevance scores in text summarization (Nenkova & McKeown, 2012). It combines two complementary frequency counts: (i) the count of the number of occurrences of a term locally within the sentence, and (ii) the count of the number of sentences in the article in which the term occurs at least once. Since relevant terms are likely to occur many times within a specific sentence but rarely in whole article, the goal of the TF-ISF statistics is to reward the terms that maximize the sentence-level count while minimizing the article-level one. The per-sentence score represented in the dataset feature is the average of the TF-ISF values of all the terms in the considered sentence.

Since the occurrences of stopwords may be accidentally correlated with a specific class label, to avoid adding noise to supervised models, text is pre-processed by removing stopwords from the list provided by the Natural Language ToolKit (Bird et al., 2009).

Sentence relevance in latent spaces. In the last years, word embedding models, such as Word2Vec (Mikolov et al., 2013) and Sent2Vec (Pagliardini et al., 2018), have become established to extract numerical features from textual data. Words in a vocabulary are represented in a latent vector space.

Word and sentence representations in latent vector spaces have been considered in this study since they are known to be able to effectively capture both syntactic and semantic similarities among text snippets (Mikolov et al., 2013). For our purposes, we model sentences as nodes of a weighted graph, where edges are weighed by the similarity score between two sentences in

the latent space. Specifically, we first generate the vector representation of a sentence using either the Word2Vec or Sent2Vec pre-trained models. Both models were pre-trained on the English-written Wikipedia corpus[1]. Then, we apply the PageRank algorithm (Page et al., 1999) on the resulting weighted graphs in order to weigh sentence relevance based on its relative authority in the graph. The generated features indicate the PageRank scores of the Word2Vec and the Sent2Vec graphs, respectively.

Sentence similarity with the abstract. Although highlights and abstracts have different purposes, they often share part of their content. Highlights usually give more result-oriented points, while abstract generally describes the position of the research work and its main contribution. However, the textual similarity between abstract and highlight content is known to be a relevant feature for highlight extraction (Collins et al., 2017).

We compute the sentence-level similarity with the abstract using the following metrics: (i) the Rouge-L score (Lin & Hovy, 2003) (described in Section 4), (ii) the average similarity between the Word2Vec sentence representations of the considered sentence and the abstract sentences, and (iii) the average similarity between the Sent2Vec representations of the considered sentence and the abstract sentences. We considered also alternative similarity measures for text (e.g. the average Kullback-Leibler divergence). However, since their integration did not produce any significant performance improvement, we will disregard them throughout the paper.

3.2. Sentence labeling

We label records in the training dataset with the maximal similarity score between the corresponding sentence and any of the article highlights. The score indicates the relevance of the sentence corresponding to the record to the article highlights. To compare the selected sentences with the reference highlights we use the standard Rouge toolkit (Lin & Hovy, 2003), which evaluates the unit overlaps between reference model and output. Among the evaluation scores provided by Rouge, we use, similar to what previously done by Collins et al. (2017), the Rouge-L F-measure, which considers the *longest common sub-sequence* of words (i.e., the largest co-occurring n-grams). Considering the largest part of the text in common between a candidate sentence

[1]The Word2Vec representation of a sentence is generated by averaging the vectorial components associated with the occurring words.

and the highlight allows us to estimate the overlap between the content of the two sentences (under the assumption that the content of the sentences selected using an extractive approach and those of the highlights are not necessarily exactly the same). A more detailed description of the Rouge evaluation scores is given in Section 4.

3.3. Model training and application

To model the correlation between sentence-level features and label values, we train a regression model on the training dataset. The model is then applied to each sentence of a not annotated (test) article. Since for each sentence of the test article a real-valued score is predicted, sentences can be directly ranked in order to generate the top-K sentence recommendation. The value K is set by the domain expert according to the number of highlights requested by the publication editor (typically between 3 and 5).

Instead of predicting the exact similarity score, an alternative strategy is to predict whether the sentence is similar or not to any of the article highlights (i.e., the binary classification problem). In such a case, to recommend the top-K sentences classifiers need to be combined with an appropriate ranking strategy (applied on the top of the classifier results). Hereafter, we will denote such an alternative strategy as the *classify-and-rank* approach.

4. Experimental results

We validated the performance of the proposed approach on three different datasets. All the experiments reported in this paper were run on a machine equipped with a Intel® Xeon® X5650, 32 GB of RAM and running Ubuntu 18.04.1 LTS. A description of the analyzed datasets, the considered algorithms, and the evaluation metrics used in the assessment procedure is given below.

Dataset. We run experiments on three different paper collections specifically designed for automatic highlight extraction. Beyond the article full-text, the collections contain the abstracts and from 3 to 6 highlights per article provided by the respective authors. Notice that, by construction, highlights are not required to exactly match any sentence of the original article.

CSPubSum is a benchmark dataset first proposed by Collins et al. (2017). It consists of 10,131 training and 150 test articles. The articles in this collection belong to the *Computer Science* field in broad terms (i.e., they range

over various subjects). To the best of our knowledge, CSPubSum (Collins et al., 2017) is the only benchmark dataset released for research purposes and tailored to automatic highlight extraction. CSPubSum was crawled from ScienceDirect (`https://www.sciencedirect.com/`), which is (to the best of our knowledge) the only scientific paper source providing both article full-text and humanly generated highlights. To extend the empirical evaluation to collections of articles ranging over specific subjects, we generated and analyzed also two new collections of papers, i.e., *AIPubSumm* and *BioPubSumm*. The subject-specific collections were crawled from ScienceDirect by performing keyword-based queries and selecting papers from the *Artificial Intelligence* and *bio-medical* domains, respectively. *AIPubSumm* consists of 198 training articles and 66 test articles, while *BioPubSumm* contains 8,070 training and 2,690 test articles. The three analyzed collections are rather different in terms of number of papers, covered topics, and text distribution.[2]

Regression and classification algorithms. We tested the performance of the proposed approach by exploiting a variety of regression and classification algorithms. Specifically, we considered the following algorithms: (i) Decision Tree Classifier (DT-C) and Regressor (DT-R). (ii) Random Forest Classifier (RF-C) and Regressor (RF-R). (iii) Multi-Layer Perceptron Classifier (MLP-C) and Regressor (MLP-R). (iv) Gradient Boosting Classifier (GB-C) and Regressor (GB-R). (v) The LSTM-based highlight extraction method proposed by Collins et al. (2017). For the approaches (i)-(iv) we used the implementation available in Scikit-Learn library (Pedregosa et al., 2011).

While the output of regression models can be directly used to rank the sentences of the input articles, classifiers perform just a selection of the potentially relevant sentences. Hence, in order to produce a ranked list of recommended sentences classifiers need to be combined with an ad hoc ranking strategy, which is applied on top of the classification process. More specifically, in the *Classify and rank* strategy the original sentences in the articles are first filtered based upon the predictions of the binary classifier (i.e., *relevant* or *not relevant*). Next, the subset of relevant sentences is ranked according to a specific ranking function in order to identify the top-K sentences. To our purpose, we tested the following ranking strategies: (i) rank sentences by decreasing PageRank score in the Sent2Vec graph (*S2V-PR*), (ii) rank sen-

[2]For the sake of reproducibility, the keys needed to crawl the subject-specific datasets are freely available at `https://git.io/Je2R0`.

tences by decreasing average tf-idf value (*Rank-Tf*), and (iii) rank sentences by decreasing Rouge-L similarity with the abstract (*Rank-Sim2Abs*).

A preliminary analysis of the impact of different ranking strategies on the performance of the classification models was carried out. In compliance with the previous findings by Collins et al. (2017), the results confirmed that the strategy (iii) was the best performing one independently of the considered algorithm. Therefore, hereafter we will consider the latter ranking strategy for the *classify-and-rank* methods.

Summarization approaches. We compared the performance of the proposed method with that of many existing summarization strategies. As unsupervised summarization methods we considered various methods based on different techniques. More specifically, the following approaches were tested: (i) LexRank (Erkan & Radev, 2004), (ii) Textrank (Mihalcea & Tarau, 2004), (iii) CoreRank (Tixier et al., 2017) (iv) LSARank (Steinberger & Jezek, 2004), and (v) MSFF (Li et al., 2012). Approaches (i)-(iii) rely on graph-based summarization strategies, which were derived from established ranking strategies, e.g., Pagerank (Page et al., 1999). Some of them have recently been applied to extract highlights from scientific articles (Collins et al., 2017). LSARank is an established summarizer relying on Latent Semantic Analysis, while MSFF is an approach based on Submodular-based technique. The considered approaches performed best in the latest summarization contests, e.g., (Giannakopoulos et al., 2015).

Since pre-trained sentence-level models based on Deep Learning Methods have recently found application in text summarization, we considered also three variants of a recently proposed summarization algorithm (Miller, 2019), which respectively rely on the following embedding models: BERT (Devlin et al., 2018), BioBERT (Lee et al., 2019) and SciBERT (Beltagy et al., 2019). BERT is among the most established sentence embedding models, while BioBERT and SciBERT are fine-tuned versions of BERT tailored to the bio-medical and scientific domains, respectively. Specializing the BERT model on a collection of scientific articles (SciBERT) allows the summarizer to better capture the semantic relationships among the contained words.

Algorithm configuration settings. To fit the models to the analyzed data distribution, we performed a grid search on the input parameters using a K-fold stratified cross validation method (Rajaraman & Ullman, 2011) (setting K to 3). Since training data are highly imbalanced, prior to training classification models we applied also oversampling techniques to re-balance data

among the two classes (i.e., *Relevant*, *Not relevant*). Table 2 summarizes the main algorithm settings. For the parameters that are not enumerated in Table 2, please consider the standard configuration setting specified in the Scikit-Learn library (Pedregosa et al., 2011) and the recommended LSTM settings given by Collins et al. (2017).

Evaluation metrics. To compare the article highlights with the sentences selected by the summarization algorithm we used the used the Rouge toolkit (Lin & Hovy, 2003). It is an established summarizer evaluation tool, which counts the unit overlaps between the two text snippets (i.e., the generated summary vs. the ground truth). In our context, we counted the overlaps between the selected top-K sentences and the expected highlights. Notice that since highlights are not necessarily part of the original articles, the matching is not required to be exact.

The Rouge scores indicate the precision, recall, and F-measure values (Rajaraman & Ullman, 2011) achieved by a summarization system according to a specific metric. To our purposes, they are tailored to the problem of recommending the K most relevant sentences. More specifically, the recall@K is the ratio of correctly selected units in the top-K sentences to all the units in expected highlights. It measures the ability to retrieve as much highlight units as possible. The precision@K is the percentage of the correctly selected units in the top-K sentences over all the units in the selected sentences. It measures the ability to accurately select relevant content. The F-measure@K is the harmonic mean of precision and recall.

We computed the recall@K, precision@K, and F-measure@K scores corresponding the following metrics: (i) *Rouge-N*: it measures the overlap of *N-grams* between the reference and automatic summaries. (ii) *Rouge-L*: it identifies the *Longest Common Subsequence* (LCS) of words between the reference and automatic summaries. Since the goal is to find the article sentences whose content is most similar to that of the highlights, Rouge-L was deemed as the most appropriate evaluation metric (Collins et al., 2017). However, for the sake of completeness, we reported also the results achieved for bigrams (N=2) and 4-grams (N=4), which are commonly used in text summarization.

Finally, to assess the ability of the proposed approach to accurately rank the article sentences based on their similarity with the reference highlights, we computed also the *Mean Reciprocal Rank* (MRR) averaged over all the reference highlights. Specifically, the MRR is computed as follows:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$$

where $rank_i$ is the ranking of the correct sentence for the i^{th} query and $|Q|$ is the total number of queries. For each highlight the MRR is computed as the mean reciprocal rank of the sentence that maximizes the Rouge-L score[3].

4.1. Comparison between classification and regression algorithms in terms of ROUGE scores

We compared the performance of various regression and classification algorithms in terms of precision, recall, and F-measure of different Rouge scores (Lin, 2004). Specifically, Figures 2-4 plot the Rouge-2, Rouge-4, and Rouge-L scores achieved on the benchmark CSPubSumm dataset by varying the number of the selected sentences. Figures 5 and 6 plot the Rouge-L scores achieved on BioPubSumm and AIPubSumm, respectively. For the sake of brevity, we omitted the corresponding Rouge-2 and Rouge-4 scores achieved on the subject-specific datasets (the achieved results were pretty similar). Within a plot, each curve corresponds to different regression or classification algorithms.

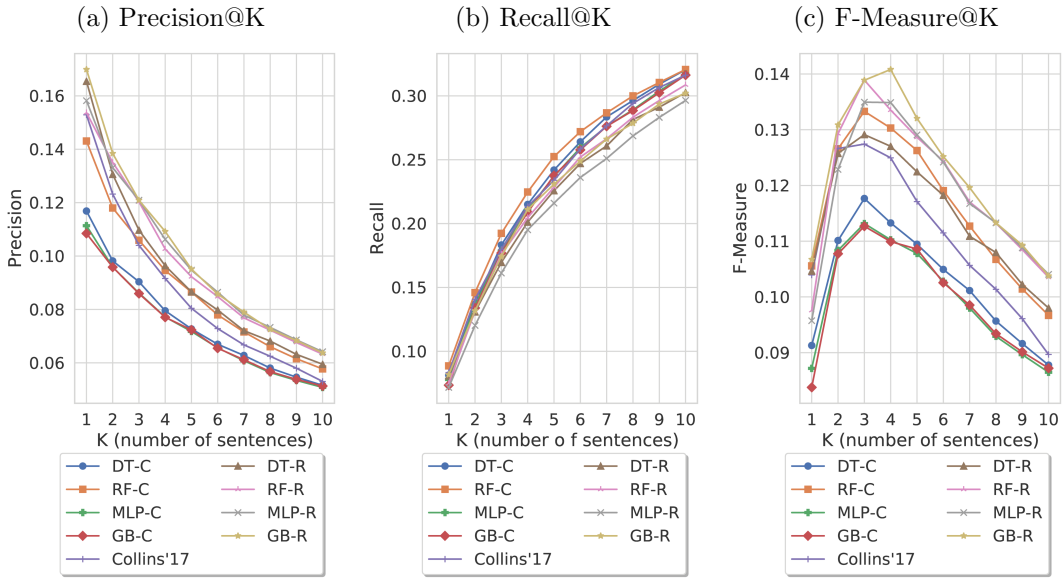
As expected, the average recall values increase while increasing the number of selected sentences, whereas the precision measures show an opposite trend. This is mainly due to the presence of redundant information when a larger number sentences is selected. Notably, the F-measure values are maximal while varying K between 2 and 5. This confirms that the selected sentences are worth considering to recommend highlight content.

Classify-and-rank methods achieved fairly high recall values because, since they tend to pick longer sentences than regression methods, in the *classify-and-rank* the selected sentences are more likely to include a larger part of relevant units. Specifically, both regression and classification models consider sentence length in the training phase. However, while regressors directly produce a ranked list of selected sentences, classifiers need a further ranking phase. Depending on the ranking functions used in the classify-and-rank strategy, sentences with variable length could be selected [4].

[3]Since highlights are unlike to be part of the article text, their content cannot be retrieved directly.

[4]The rationale is to recommend the most appropriate article sentences (independently

Figure 2: CSPubSumm dataset: comparison between classification and regression methods in terms of Rouge-2 scores.

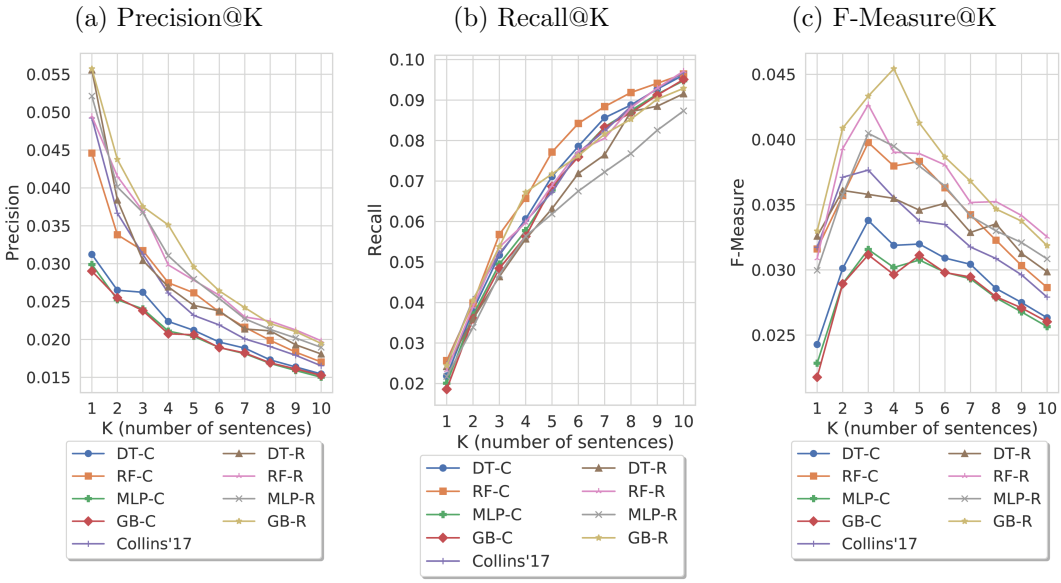


The precision values achieved by the classify-and-rank strategies were fairly low, because these sentences include redundant content as well. Conversely, regression methods performed best in terms of precision and F-measure. For example, by setting K to 3 the best performing regressor (Gradient Boosting Regressor) achieved a Rouge-L F-measure equal to 0.316 against 0.298 achieved the best competitor (i.e., the classify-and-rank method based on Random Forest and Sim2Abs). Regression methods pick shorter yet relevant sentences, which are deemed as more appropriate to be recommended as candidate highlights (typically, due to editorial constraints, article highlights cannot exceed a maximal length). Notice that sentence length is an input feature, which can be considered by the regression and classification models in order to make appropriate sentence recommendations.

To give more insights into the achieved results, Table 3 reports the F-measure results of all the tested methods (including the summarization methods, whose results will be thoroughly discussed in Section 4.2) for three rep-

of the eventual length constraint enforced by the publisher) to experts in order to support the article annotation process.

Figure 3: CSPubSumm dataset: comparison between classification and regression methods in terms of Rouge-4 scores.



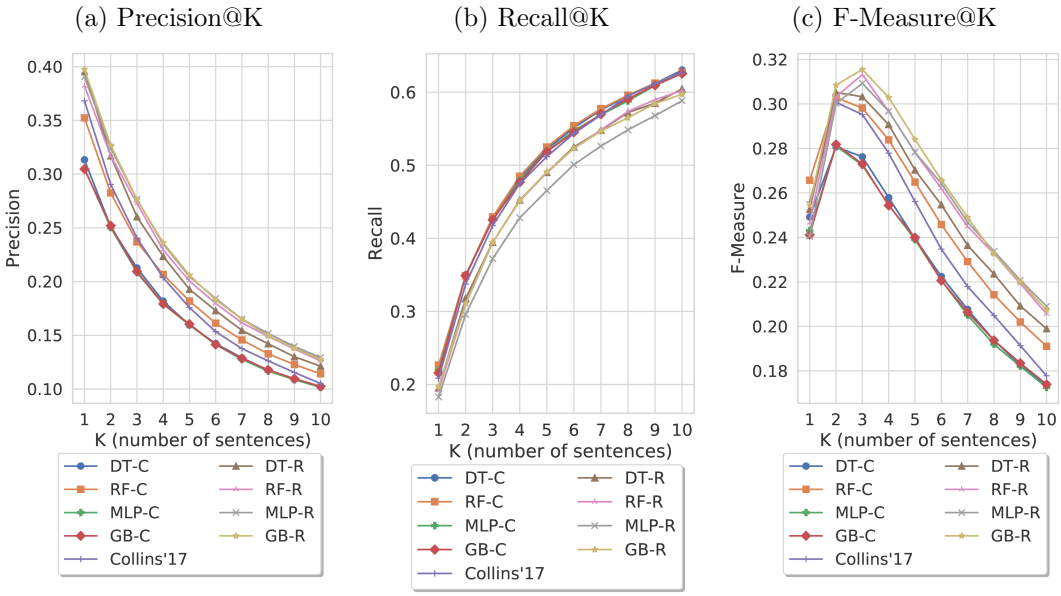
resentative K values. To assess the significance of the achieved performance improvements, we applied the t-test validation with 95% confidence level. Significant variations, in terms of Rouge-L F-measure, between the best regression method (GB Regressor) and the other algorithms are starred in Table 3. By varying K in the range between 3 and 5 the performance improvements achieved by the regression method against the largest majority of the classify-and-rank and summarization approaches are statistically significant.

4.2. Comparison with summarization methods in terms of ROUGE scores

We compared the performance of the best regression and classification methods with that of various summarization algorithms. Despite the goal of the proposed approach (i.e., extract article highlights) significantly differs from those of traditional summarization methods (i.e., extract a generic summary of the article content), we investigated to what extent existing summarization algorithms could be exploited to address the specific research task tackled by this work.

Figure 7 shows the results of a performance comparison, conducted on the

Figure 4: CSPubSumm dataset: comparison between classification and regression methods in terms of Rouge-L scores.



benchmark CSPubSumm dataset, between the most popular summarizers, the Deep Learning methods based on the BERT embedding model, and the best regressor (Gradient Boosting Regressor) and classify-and-rank method (Random Forest Classifier). Figure 8 summarizes similar results achieved on the subject-specific datasets.

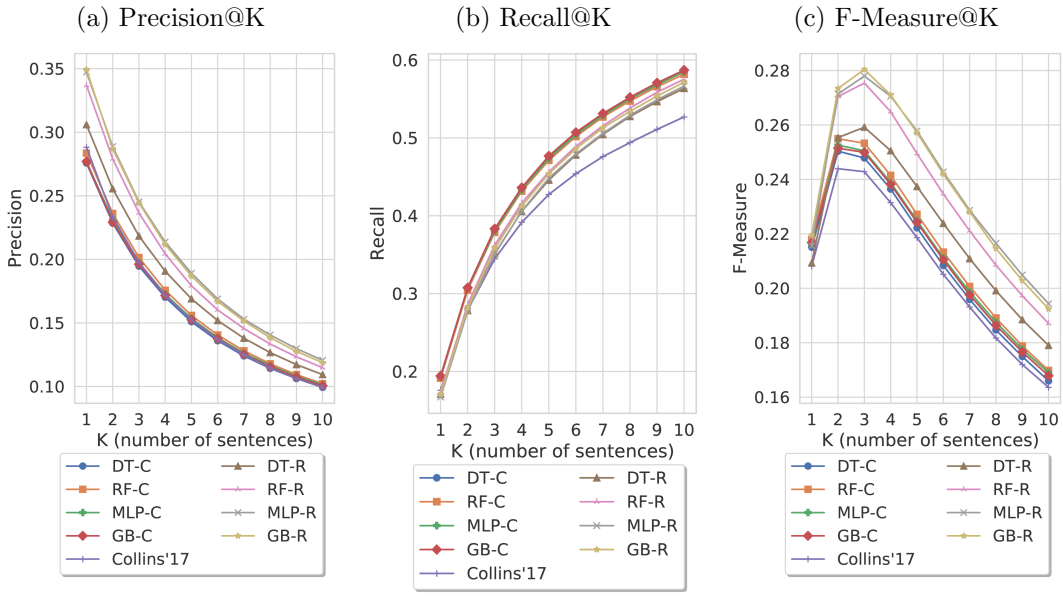
The achieved results confirm that the proposed solutions based on regression models is superior to all the traditional and BERT-based summarization approaches, especially when the number of selected sentences in the range between 3 and 5 (i.e., the most common number of requested highlights).

4.3. Comparison with the abstract

We made also an attempt to use the content of the abstract of the paper as a baseline for highlight extraction. The performance of the baseline was significantly worse than that achieved using regression-based techniques (e.g., on the CSPubSumm dataset GB regression 0.316 vs. baseline 0.284 with k=3, 0.303 vs. 0.282 with k=4 in terms of Rouge-L F-measure).

Notice that despite the abstract could be deemed as a paper summary as itself, its scope completely differs from those of the highlights. In fact,

Figure 5: BioPubSumm dataset: comparison between classification and regression methods in terms of Rouge-L scores.



highlights are mainly result-oriented whereas the abstract is general-purpose. Therefore, using the abstract content to recommend highlights could be misleading. Furthermore, publishers often make the abstract accessible along with the highlights. Hence, their content should not be overlapped.

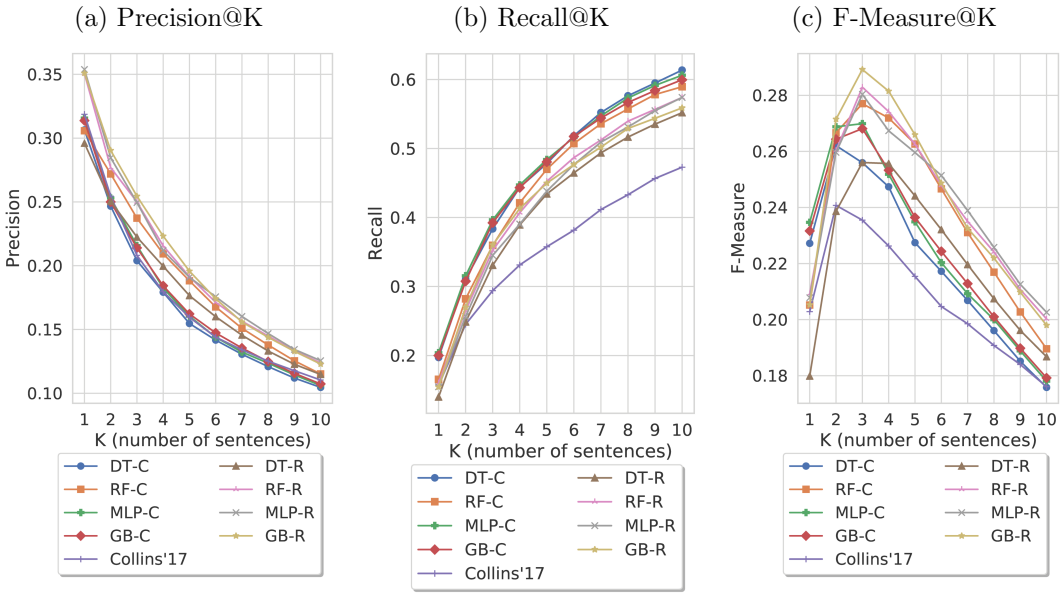
4.4. Comparison with the other methods in terms of Mean Reciprocal Rank

We compared the sentence ranks produced by all the tested methods using the MRR measure. This metric has been evaluated for three datasets under test. Specifically, Figure 9, 10 and 11 report the MRR results respectively for CSPubSumm, BioPubSumm and AIPubSumm.

These comparisons allow us to quantitatively evaluate the ability of the regression model to rank sentences according to their pertinence to the actual highlight content.

The results show that regression methods perform significantly better than the other approaches, since they are able to better discriminate sentences according to their actual relevance to the highlight content.

Figure 6: AIPubSumm dataset: comparison between classification and regression methods in terms of Rouge-L scores.



4.5. Execution times

The feature extraction and data labelling steps approximately took 25 seconds per article, while the model training time varied between 76 seconds (Decision Tree Classifier) and 59 minutes (Gradient Boosting Regressor) on the CSPubSumm dataset.

5. Conclusions and future works

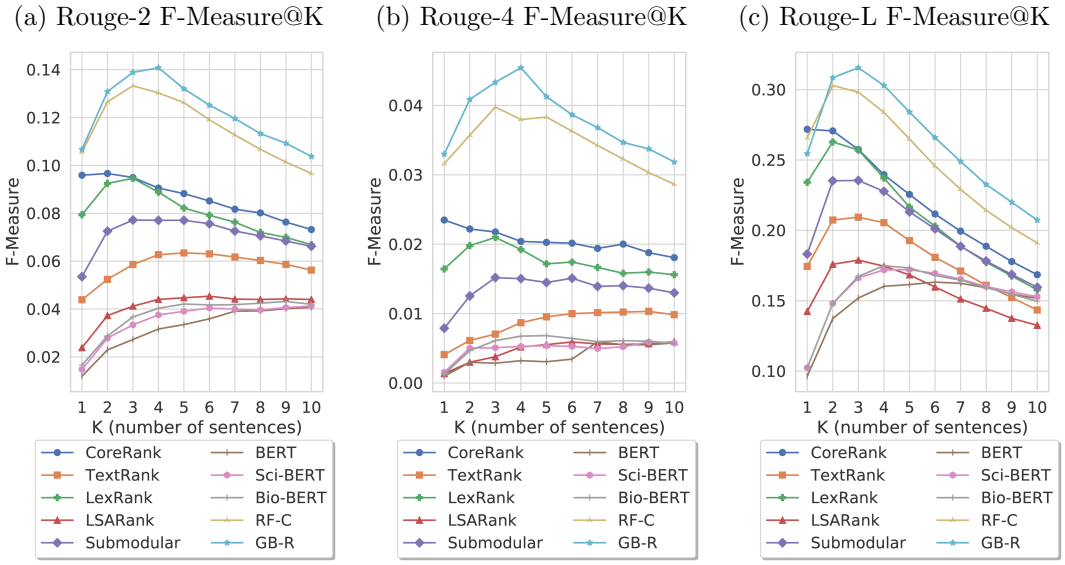
Annotating scientific articles with textual highlights provides readers with potentially useful result-oriented insights. Unfortunately, the annotation process is commonly performed manually. Furthermore, for most of the past publications highlight information is missing. This work overcomes the aforesaid issues by performing supervised learning on previously annotated articles.

The results achieved on the benchmark CSPubSumm dataset and on the two new subject-specific collections show that (i) regression models, in most cases, performed significantly better than all classification models, including the state-of-the-art approach proposed by Collins et al. (2017). (ii) The

performance of the proposed approach was superior to that of unsupervised summarization methods, which have shown to be unsuitable for addressing the specific highlight extraction task. (iii) The features denoting the similarity between candidate sentences and the highlights in the vector spaces of word/sentence embeddings appearing to be influential for sentence selection and ranking. (iii) Considering the similarity between the sentence and the abstract helps to train effective models as well. However, abstract sentences as themselves appeared to be inappropriate as paper highlights. (iv) While coping with homogeneous paper collections (i.e., papers ranging on the same topic), the extraction process becomes less sensitive to the presence of outliers. Therefore, the performance of classification models, in some cases, gets closer to that of best performing regressors.

The above-mentioned research findings leave room for various extensions on the current work. First, regression models can be better customized on specific topics, e.g., by combining fine-tuned embedding models with a number of descriptive features. Secondly, a similar regression method can be applied to the phrases of the article text in order to automatically extract alternative keyphrases or titles. Finally, the correlation between keywords,

Figure 7: CSPubSumm dataset: performance comparison between the best performing regression and classification methods, the most popular summarization methods and summarization methods based on BERT.



highlights, title, and abstracts can be further explored. Specifically, we plan to design an integrated model extracting keywords, highlights, abstracts and titles from scientific articles with coherent, exhaustive, and non-redundant content. The designed model may drive content extraction from both single articles and collections of homogeneous articles.

References

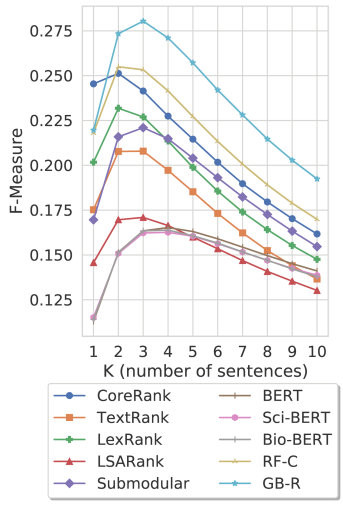
Alguliyev, R. M., Aliguliyev, R. M., Isazade, N. R., Abdi, A., & Idris, N. (2019). Cosum: Text summarization based on clustering and optimization. *Expert Systems*, *36*, e12340.

Baralis, E., Cagliero, L., Fiori, A., & Garza, P. (2015). Mwi-sum: A multilingual summarizer based on frequent weighted itemsets. *ACM Trans. Inf. Syst.*, *34*, 5:1–5:35. URL: `https://doi.org/10.1145/2809786`. doi:`10.1145/2809786`.

Beltagy, I., Cohan, A., & Lo, K. (2019). Scibert: Pretrained contextualized embeddings for scientific text. *arXiv preprint arXiv:1903.10676*, .

Figure 8: BioPubSumm and AIPubSumm datasets: performance comparison between the best performing regression and classification methods, the most popular summarization methods and summarization methods based on BERT.

(a) (BioPubSumm) Rouge-L F-Measure@K



(b) (AIPubSumm) Rouge-L F-Measure@K

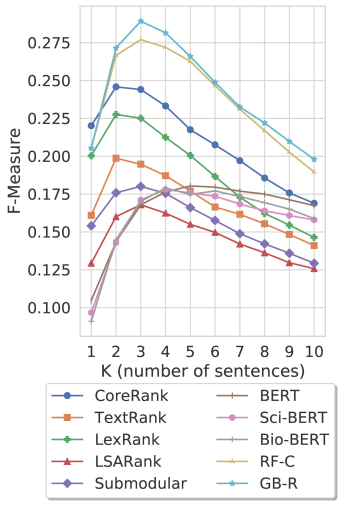


Figure 9: CSPubSumm dataset: Mean Reciprocal Ranks achieved by regression, classification, and summarization methods

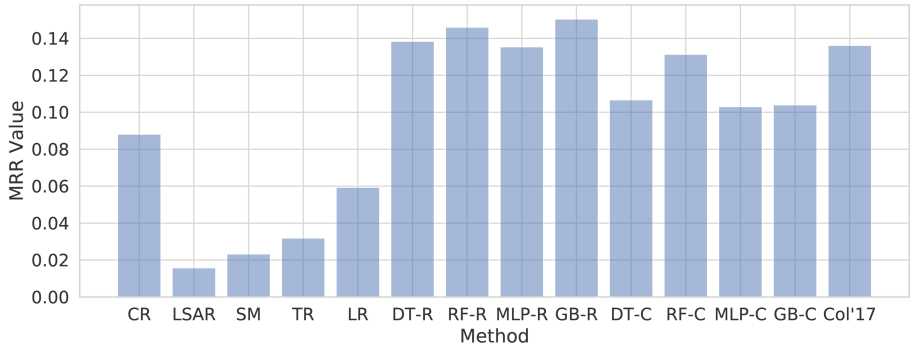
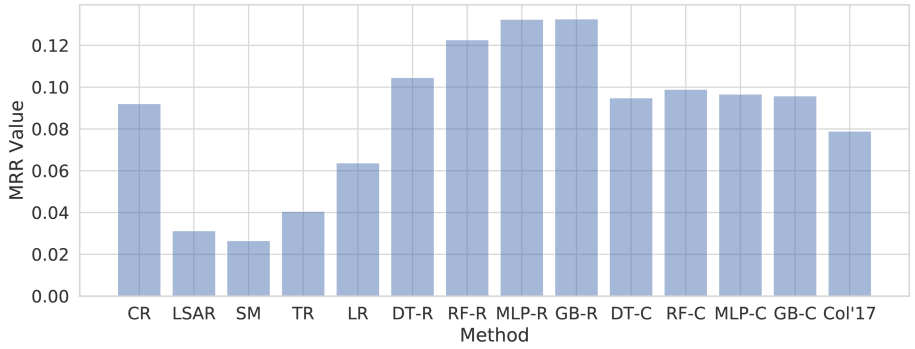


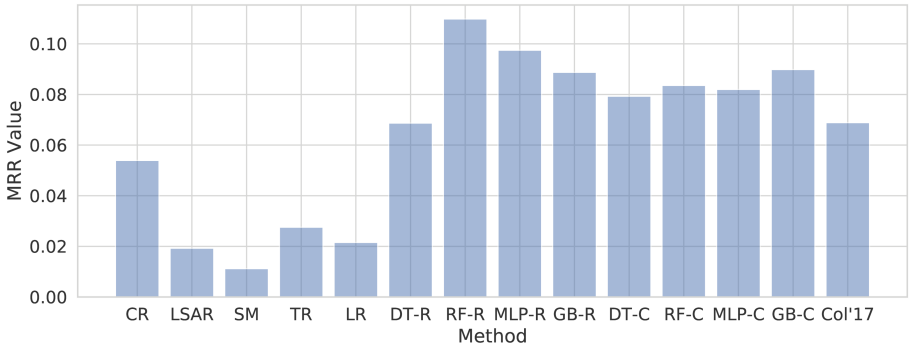
Figure 10: BioPubSummm dataset: Mean Reciprocal Ranks achieved by regression, classification, and summarization methods



Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Cagliero, L., Garza, P., & Baralis, E. (2019). Elsa: A multilingual document summarization algorithm based on frequent itemsets and latent semantic analysis. *ACM Trans. Inf. Syst.*, *37*, 21:1–21:33. URL: `http://doi.acm.org/10.1145/3298987`. doi:`10.1145/3298987`.

Figure 11: AIPubSumm dataset: Mean Reciprocal Ranks achieved by regression, classification, and summarization methods



Collins, E., Augenstein, I., & Riedel, S. (2017). A supervised approach to extractive summarisation of scientific papers. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)* (pp. 195–205).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, .

Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, *22*, 457–479.

Fang, C., Mu, D., Deng, Z., & Wu, Z. (2017). Word-sentence co-ranking for automatic extractive text summarization. *Expert Systems with Applications*, *72*, 189–195.

Giannakopoulos, G. (2013). Multi-document multilingual summarization and evaluation tracks in acl 2013 multiling workshop. In *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization* (pp. 20–28). Association for Computational Linguistics. URL: `http://www.aclweb.org/anthology/W13-3103`.

Giannakopoulos, G., Kubina, J., Conroy, J., Steinberger, J., Favre, B., Kabadjov, M., Kruschwitz, U., & Poesio, M. (2015). Multiling 2015: Mul-

tilingual summarization of single and multi-documents, on-line fora, and call-center conversations, . doi:10.18653/v1/W15-4638.

Gillick, D., & Favre, B. (2009). A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing* (pp. 10–18). Boulder, Colorado: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W09-1802.

Gollapalli, S. D., & Caragea, C. (2014). Extracting keyphrases from research papers using citation networks. In *AAAI*.

Hasan, K. S., & Ng, V. (2014). Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (pp. 1262–1273). volume 1.

Kim, M., Moirangthem, D. S., & Lee, M. (2016). Towards abstraction from extraction: Multiple timescale gated recurrent unit for summarization. In *Rep4NLP@ACL* (pp. 70–77). Association for Computational Linguistics.

Krapivin, M., Autayeu, A., Marchese, M., Blanzieri, E., & Segata, N. (2010). Keyphrases extraction from scientific documents: improving machine learning approaches with natural language processing. In *International Conference on Asian Digital Libraries* (pp. 102–111). Springer.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2019). Biobert: pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746*, .

Li, J., Li, L., & Li, T. (2012). Multi-document summarization via submodularity. *Applied Intelligence*, *37*, 420–430.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, .

Lin, C.-Y., & Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1* (pp. 71–78).

Litvak, M., Last, M., & Friedman, M. (2010). A new approach to improving multilingual summarization using a genetic algorithm. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (pp. 927–936). Uppsala, Sweden: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/P10-1095.

Litvak, M., Last, M., & Vanetik, N. (2015). Krimping texts for better summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1931–1935).

Lloret, E., Rom-Ferri, M. T., & Palomar, M. (2013). Compendium: A text summarization system for generating abstracts of research papers. *Data & Knowledge Engineering*, *88*, 164 – 175. doi:https://doi.org/10.1016/j.datak.2013.08.005.

Mao, X., Yang, H., Huang, S., Liu, Y., & Li, R. (2019). Extractive summarization using supervised and unsupervised learning. *Expert Systems with Applications*, *133*, 173–181.

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Miller, D. (2019). Leveraging bert for extractive text summarization on lectures. *arXiv preprint arXiv:1906.04165*, .

Naik, A. P., & Bojewar, S. (2017). Tweet analytics and tweet summarization using graph mining. In *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)* (pp. 17–21). volume 1. doi:10.1109/ICECA.2017.8203674.

Nallapati, R., Zhai, F., & Zhou, B. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. In *Mining text data* (pp. 43–76). Springer.

Nikolov, N. I., Pfeiffer, M., & Hahnloser, R. H. (2018). Data-driven summarization of scientific articles. In *Proc. of the 7th International Workshop on Mining Scientific Publications, LREC 2018*.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The PageRank citation ranking: Bringing order to the web.*. Technical Report Stanford InfoLab.

Pagliardini, M., Gupta, P., & Jaggi, M. (2018). Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 528–540). volume 1.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Rajaraman, A., & Ullman, J. D. (2011). *Mining of Massive Datasets*. New York, NY, USA: Cambridge University Press.

Steinberger, J. (2013). The UWB summariser at multiling-2013. In *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization* (pp. 50–54). Sofia, Bulgaria: Association for Computational Linguistics. URL: `https://www.aclweb.org/anthology/W13-3107`.

Steinberger, J., & Jezek, K. (2004). Using latent semantic analysis in text summarization and summary evaluation. *Proc. ISIM*, *4*, 93–100.

Steinberger, J., Kabadjov, M., Steinberger, R., Tanev, H., Turchi, M., & Zavarella, V. (2011). JRC's participation at TAC 2011: Guided and multilingual summarization tasks. In *TAC'11: Proceedings of the The 2011 Text Analysis Conference*.

Takamura, H., & Okumura, M. (2009). Text summarization model based on maximum coverage problem and its variant. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)* (pp. 781–789). Athens, Greece: Association for Computational Linguistics. URL: `https://www.aclweb.org/anthology/E09-1089`.

Tixier, A., Meladianos, P., & Vazirgiannis, M. (2017). Combining graph degeneracy and submodularity for unsupervised extractive summarization. In *Proceedings of the Workshop on New Frontiers in Summarization* (pp. 48–58).

Wong, K.-F., Wu, M., & Li, W. (2008). Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1* (pp. 985–992). Association for Computational Linguistics.

Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., & Zhao, T. (2018). Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 654–663).

Table 1: Examples of abstract, highlights, and recommended sentences. Paper entitled *Context adaptable driver information - Or, what do whom need and want when?* in the *CSPubSumm* Collins et al. (2017) dataset.

Qualitative assessment of the recommended sentences	
Abstract	This study deals with a first step towards context adaptive functionality of a Driver Information System. Driving a car is a complex task for which the driver needs appropriate information to fulfil his or her goals. New technologies enable adaptability to driver state, task, personality etcetera and also to the context. The aim of this study was therefore to investigate what information people perceive that they need and want from the car in different contexts and to what extent there is consensus about the function. A new methodology was developed, and 33 private car drivers were interviewed and asked to rate a number of possible abstract functions in a car in different contexts. It was shown that people need and want different types of information in different contexts. It was furthermore indicated that there is sometimes a difference in drivers' opinions about what should be presented by the car and that there is varying consensus over different functions in different contexts. The rating result was illustrated by an easily perceived Context Function Matrix. The results may be used in the design of a context adaptive driver information system.
Manually annotated highlights	As guideline for design of context adaptive driver information systems or for optimization of display space.
	As a weight when evaluating future adaptive information systems.
	When deciding whether a function should be activated automatically or manually.
Gradient Boosting Regressor (Best algorithm)	The study resulted in a context function matrix and a zoom metaphor useful for future context adaptive driver information.
	Not surprisingly, the results indicate that drivers want or need different functions in different contexts.
	The results can be used as a guideline for design of context adaptive driver information systems or for optimization of display space.
Random Forest Classifier (Ranking: Sim-Abs score)	The purpose of this study was therefore to investigate: whether drivers want or have a perceived need for different functions in different contexts (Q1), what information different drivers perceive to be needed and wanted in different contexts (Q2), the extent to which there is consensus about each function (Q3) And to illustrate and make understandable the functions in the different contexts (Q4).
	The first research question was whether drivers want or have a perceived need for different functions in different contexts (Q1), The interviews, function grading and open end answers in the study gave an indication that drivers have different perceived needs and desires in different driving contexts.
	For instance, a tired driver and an alert driver, a daily trip to work and a holiday trip, a worn car and a new car and drivers with long and short response times may need different information.
CoreRank (Graph-based method)	There was a high consensus about the lowest grades: show engine coolant temperature, show oil level in engine, show engine oil temperature, measure time, show travel distance in total and show engine oil pressure. Lap time, show cruise control set speed, ability to watch movie, show when it is permitted to take over, show engine oil temperature and show average speed were given the lowest scores but had a high consensus.
	Before driving, functions of a more strategic character are graded high: warn for slippery road conditions, show outdoor temperature, warn for slippery road conditions on the way to the destination, show fuel level, show distance to empty tank, show alternative roads to the destination, show information about dangerous roads, show estimated time of arrival, show that there are queues on the way to the destination, show recommended speed due to road conditions, visibility and show tire pressure in the different tires.
	Ability to surf on the Internet, show free parking places, show engine oil temperature, show engine oil pressure, lap time, show start time for parking heater and remind that the car needs regular service received the lowest grades.

Table 2: Input parameters of regression and classification algorithms.

Dataset	**Model**	**Parameters**
CSPubSumm	DT-R	Criterion: Friedman MSE, max depth: 8
	RF-R	Criterion: MSE, max depth: 10, estimators=100
	MLP-R	Number of layers: 4, layer size: 60
	GB-R	Loss: least squares, learning rate: 0.18, estimators: 400
	DT-C	Criterion: gini, max depth: 8
	RF-C	Criterion: entropy, max depth: 20, estimators:80
	MLP-C	Number of layers: 1, layer size: 60
	GB-C	Loss: deviance, learning rate: 0.18, estimators: 400

Table 3: Rouge-L F-Measure values achieved by different algorithms. Significant variations between the best regression method (GB regressor) and the other algorithms are starred.

K	**Core-Rank**	**LSA-Rank**	**Sub-Modul**	**Text-Rank**	**Lex-Rank**	**DT Reg**	**RF Reg**	**MLP Reg**	**GB Reg**	**DT Cla**	**RF Cla**	**MLP Cla**	**GB Cla**	**LSTM Cla**
Original CSPubSumm dataset														
3	0.257*	0.179*	0.235*	0.209*	0.257*	0.303*	0.313	0.309*	**0.316**	0.276*	0.298	0.272*	0.273*	0.295
4	0.239*	0.175*	0.228*	0.205*	0.237*	0.291*	0.297*	0.297*	**0.303**	0.258*	0.284*	0.254*	0.254*	0.278*
5	0.225*	0.168*	0.213*	0.193*	0.217*	0.270*	0.278*	0.278*	**0.284**	0.239*	0.265*	0.239*	0.240*	0.256*
BioPubSumm Dataset														
3	0.241*	0.171*	0.221*	0.208*	0.227*	0.259*	0.275*	0.278	**0.280**	0.248*	0.253*	0.250*	0.250*	0.243*
4	0.227*	0.166*	0.215*	0.197*	0.223*	0.250*	0.265*	0.270	**0.271**	0.236*	0.241*	0.239*	0.238*	0.231*
5	0.215*	0.160*	0.204*	0.185*	0.199*	0.237*	0.249*	**0.258**	0.257	0.222*	0.227*	0.225*	0.224*	0.219*
AIPubSumm Dataset														
3	0.244*	0.168*	0.180*	0.195*	0.225*	0.256	0.283	0.280	**0.289**	0.256	0.277	0.270	0.268	0.235*
4	0.233*	0.162*	0.175*	0.187*	0.212*	0.256	0.274	0.267	**0.281**	0.247	0.271	0.252	0.253	0.226*
5	0.217*	0.155*	0.166*	0.177	0.201*	0.244	0.263	0.259	**0.266**	0.227*	0.263	0.235	0.236	0.215*