CIT. Journal of Computing and Information Technology, Vol. 28, No. 3, September 2020, 183–205
doi: 10.20532/cit.2020.1005160

A Survey of Citation Recommendation Tasks and Methods

Zoran Medić and Jan Šnajder

Text Analysis and Knowledge Engineering Lab, Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia

Scientific articles store vast amounts of knowledge amassed through many decades of research. They serve to communicate research results among scientists but also for learning and tracking progress in the field. However, scientific production has risen to levels that make it difficult even for experts to keep up with work in their field. As a remedy, specialized search engines are being deployed, incorporating novel natural language processing and machine learning methods. The task of citation recommendation, in particular, has attracted much interest as it holds promise for improving the quality of scientific production. In this paper, we present the state-of-the-art in citation recommendation: we survey the methods for global and local approaches to the task, the evaluation setups and datasets, and the most successful machine learning models. In addition, we overview two tasks complementary to citation recommendation: extraction of key aspects and entities from articles and citation function classification. With this survey, we hope to provide the ground for understanding current efforts and stimulate further research in this exciting and promising field.

ACM CCS (2012) Classification: Computing methodologies → Artificial intelligence → Natural language processing

Keywords: scientific articles, scientific text processing, machine learning, deep learning, natural language processing, citation recommendation

1. Introduction

Scientific articles are considered one of the main research resources in today's world. The first published scientific article dates back as far as 1665 when the Royal Society[1] published the first volume of Philosophical Transactions of the Royal Society [45]. Since then, scientists have published millions of articles describing their research ideas and findings. Articles are used not only to communicate research results among fellow scientists but also as the key resources for learning and tracking progress in the field. Without a doubt – and despite the many diverse sources of scientific information now available online – scientific articles still play a major role in storing and disseminating human knowledge and will likely continue doing so in the foreseeable future. Additionally, technological advances and the rise of the Internet have accelerated science and scientific production to unprecedented levels. While the benefits for society are undeniable, the downside of modern science is that the number of published scientific articles has recently been growing to the extent that scientists are finding it difficult to keep up with published research. A recent study reports that the number of science and engineering articles that were published between 2004 and 2014 grew at an average annual rate of 6%, reaching almost 2.3 million in 2014 [69]. Such a growth rate puts a strain on scientists to become more selective and filter the articles they would like to read, as reading all the relevant publications becomes infeasible. A study published in [64] found that in 2012, scientists from US and Australian universities estimated that they read an average of 22 articles per month, which matches the number reported in an earlier study from 2005. These findings suggest that scientists may have already reached the performance

[1] https://royalsocietypublishing.org

ceiling in regard to the number of articles they can process within a fixed time frame.

Technology has contributed to the overwhelming increase in scientific publishing, but it can also offer remedies. In particular, specialized search engines, indexing large scientific databases, can facilitate access to published articles for scientists and help them make the best use of their time to keep track of progress in their field. A number of such search engines are in widespread use today, including Google Scholar[2], Microsoft Academic[3], and Semantic Scholar[4]. These systems index various pieces of information extracted either from article text or its metadata, including keywords, author data, publication data, and citations, making it possible for users not only to retrieve the most relevant articles for their query but also to semantically navigate through entire collections of articles as well as to recommend articles for reading. With recent staggering advances in artificial intelligence and machine learning, in particular *natural language processing* (NLP), search engines have begun incorporating more sophisticated techniques for semantic processing of scientific articles. For example, Semantic Scholar now uses a machine learning model that identifies which citations in a given scientific article were most influential for that article [2], a feature that might help scientists find relevant articles more easily. A number of other NLP methods are being used to that end, including extraction of key phrases or key aspects from articles [22, 35], argumentation mining over article sentences [60, 19, 37], classification of citations into different categories [1, 62], article summarization [50, 11], and citation recommendation [5, 24, 28]. The most successful systems leverage the textual content of the articles in combination with various metadata obtained from the citation network (a graph linking papers and authors who cite each other).

In this article, we focus on arguably one of the most exciting aspect of scientific text processing – *citation recommendation*. Citation recommendation is the task of automatically identifying, from a collection of scientific articles, an article that could or should have been cited in another article or that may be cited in a yet unpublished manuscript. Solving this task has the potential to directly improve the quality of scientific production, as it can ensure that all relevant precursory work has been identified and properly contextualized. We provide an overview of the citation recommendation task and a survey of the research and practices in the field, focusing in particular on the recent advances made possible with deep learning and neural NLP. We recognize that a fullfledged citation recommendation system should be capable of capturing not only the information about key aspects of the recommended article (*i.e.*, cited article) but also of the article or manuscript that the recommendation is being made for (*i.e.*, *citing* article). Moreover, a welcome feature would be for the system to provide reason for citing an article – information that is available within the snippet of the text in which a citation occurred (*i.e.*, *citation* context). Considering this, we include in our survey two other tasks in scientific text processing – *extraction of key aspects* from scientific articles and *citation function classification* – both of which can serve to improve the performance of citation recommendation systems. In principle, a system capable of extracting article's key aspects and determining the function of citation should be able to use this information to produce more accurate but also more comprehensible citations, offering detailed information about the cited articles and the reasons for citing them. We start this survey with an overview of the two supporting tasks, followed by an overview of the models used in the citation recommendation tasks. The survey is intended for readers with some background in machine learning and natural language processing. While we have tried to cover the most relevant and recent work, we have no claim to completeness, and the interested readers are encouraged to follow the references for a deeper understanding of the topics. Table 1 lists the articles reviewed in this work and organized per tasks.

The rest of the article is organized as follows. Section 2 introduces some of the common approaches in the task of extraction of key aspects

[2]https://scholar.google.com

[3]https://academic.microsoft.com

[4]https://www.semanticscholar.org

Table 1. Overview of articles covered in our work, organized per task and listed in the order of publishing.

Task	Reviewed articles
Extraction of key aspects	Guo *et al.* [21] (2010) Gupta and Manning [22] (2011) Heffernan and Teufel [25] (2018)
Extraction of entities	Luan *et al.* [42] (2018) Jain *et al.* [30] (2020)
Citation function classification	Teufel *et al.* [62] (2006) Abu-Jbara *et al.* [1] (2013) Jurgens *et al.* [33] (2018) Cohan *et al.* [12] (2019) Beltagy *et al.* [4] (2019)
Global citation recommendation	Bethard and Jurafsky [5] (2010) Ren *et al.* [52] (2014) Bhagavatula *et al.* [6] (2018) Cohan *et al.* [13] (2020)
Local citation recommendation	He *et al.* [24] (2010) Huang *et al.* [28] (2015) Ebesu and Fang [16] (2017) Yang *et al.* [70] (2019)

and entities from scientific articles. Section 3 provides descriptions of various models used for the task of citation function classification. Section 4 describes the citation recommendation task, outlining the difference between the two approaches (global and local), with an overview of available datasets, metrics used for evaluation of the models, and descriptions of the current state-of-the-art models in the field. Section 5 concludes the paper with a number of research ideas for future work.

2. Extraction of Key Aspects and Entities

Scientific writing typically differs from the writing styles of other text genres. In many research areas, a typical analytical or experimental scientific article will utilize the common pattern of first introducing the background and related work, then defining the problem, proposing a solution for it, and evaluating it. These patterns emphasize the so-called key aspects [22] of an article, which in effect define the information structure of the text [21]. Along with key aspects, another information bearing element of text are the different entities that are being defined, brought into relation, or otherwise mentioned in the article. The entity types differ from those types commonly encountered in standard NLP (*e.g.*, in processing news articles), and depend on the particular scientific domain. For instance, in experimental papers in the field of artificial intelligence, the entities will usually include different datasets, tasks, metrics, *etc.*

Detection and extraction of such aspects and entities from scientific articles have been the focus of much recent research. Approaches to key aspect extraction differ with respect to the granularity of patterns extracted, in that some approaches segment sentences in the articles into predefined aspects, while others extract phrases and keywords from the articles that best describe certain aspects. Similarly, approaches to entity extraction differ in whether the entities are extracted from article abstracts or from the entire text of the article. We next review some of the more prominent papers focusing on the extraction of key aspects and entities.

2.1. Extraction of Key Aspects

Guo *et al.* [21] evaluated three different categorization schemes for key aspect extraction on a set of abstracts from the biomedical domain: (1) section names [26], in which abstract sentences are subcategorized into objective, method, the results, and conclusion types, (2) argumentative zoning [61], from which they filter out seven categories that do appear in abstracts, and (3) Core Scientific Concepts [40], a finegrained annotation scheme, containing 10 categories. The authors annotated 1,000 abstracts at the sentence level using all three schemes and used a machine learning model to categorize sentences into the corresponding categories. Annotated dataset was rather imbalanced for all three schemes, with sentences containing the results aspects prevailing in all schemes. As features, they used a combination of lexical (unigram and bigram counts, part-of-speech tags, *etc.*) and positional features (location of a sentence in the abstract, category of previous sentence). The best-performing model was a Support Vector Machine (SVM) with linear kernel, reaching over 80% in accuracy for all three categorization schemes. This demonstrated that aspects from all three schemes can be automatically extracted from scientific articles with a satisfactory accuracy, although a decrease in accuracy was observed for the finegrained scheme, suggesting that more training data may be needed.

Gupta and Manning [22] carried out a similar annotation task on a set of articles from the domain of computational linguistics. They extracted three types of key aspects from article's title and abstract: focus (article's main contribution), domain (article's application domain), and the techniques used (the method or the tool used in the experiments). Contrary to the work of Guo *et al.* [21], they did not require the extracted aspects to span entire sentences, but rather extracted only the phrases describing certain aspects. The annotated corpus totals 474 abstracts annotated with three key aspect types. Instead of training a machine learning model on the annotated corpus and applying the model to a new set of articles, the authors used a bootstrapping approach to expand the set of patterns used for expressing each aspect in the annotated dataset. An evaluation showed that bootstrapping can extract key aspects with a quality that is comparable to that obtained with costly and tedious human annotation. As follow-up research, they used the obtained patterns for extracting key aspects from a set of articles published at various venues of the *Association for Computational Linguistics* (ACL)[5], dating from 1965 to 2009, and analyzed how certain subfields in computational linguistics evolved over time.

In a different approach, Heffernan and Teufel [25] introduced the task of identifying two key aspects in scientific articles: problems and solutions. Contrary to previous work, the authors decided to focus on these two key aspects only, motivated by the fact that research is often described as a problem-solving activity and article's text should therefore contain descriptions of both the problems and their solutions. The authors frame the task as two binary classification tasks: one for each key aspect, with negative examples being those phrases that are neither problem nor solution descriptions. To obtain positive examples for training the classifier, they extract sentences containing problem and solution-bearing words. Those words were obtained using the word2vec algorithm [44], which creates vector representations for words and enforces higher similarity between vectors of words that often appear close to each other. Such enforcing leads to semantically similar words having similar vector representations obtained with word2vec algorithm, which the authors use for detecting synonyms of "problem" and "solution". Sentences that do not contain problem or solution-bearing words are used as negative examples in classification tasks, with the final dataset in the end containing an equal number of positive and negative examples for both classification tasks. Finally, two binary classifiers are trained: one for problem and another for solution detection. For both tasks, the best results were obtained using an SVM with a combination of lexical (*e.g.*, bag-of-words, word polarity, and part-of-speech) and embedding (word2vec and doc2vec [38]) features, with both models achieving over 80% accuracy. Although the results suggest that the identification of problem and solution descriptions is possible when these are limited to one-sentence

[5]https://www.aclweb.org

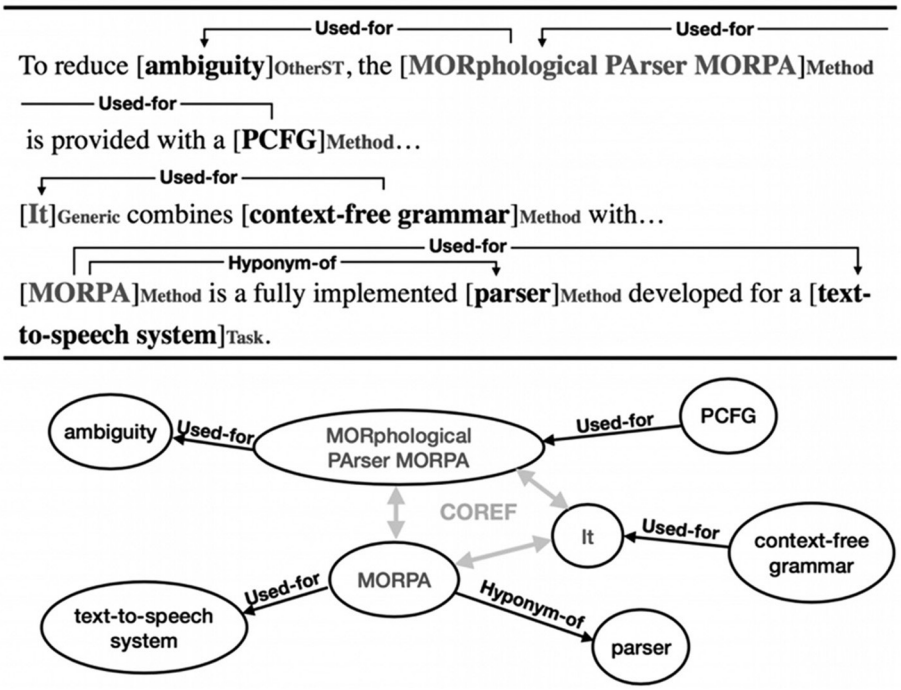

Figure 1. An example of annotation from the SciERC dataset. The figure shows entities (printed in bold) and relations among them (indicated with arrows containing the relation name), together with a sample graph constructed from entities and relations from the example. The sample graph shows coreference links between the extracted entities (in light gray color) that were extracted from the text displayed in the upper part of the figure. (Reprinted from [42] under Creative Commons-BY-4.0 license.)

spans, in reality the problem and solution descriptions typically span a number of sentences, and the approach would have to be extended to account for this. Additionally, it would be interesting to see whether joint detection of problems and solutions would yield better results, as these two aspects of an article intuitively appear to be related.

2.2. Extraction of Entities

In this subsection, we review a line of work that focuses on the extraction of smaller information units in scientific texts – entities. Extraction of entities provides a more detailed understanding of the information presented in articles through a variety of entity types and relations that can be detected among those entities. Here, we briefly review two recent datasets that focus on extraction of entities from scientific articles.

Luan *et al.* [42] introduced SciERC, a dataset of 500 abstracts annotated with (1) entities (defined as single or multiword phrases extracted from the article's text), (2) relations between the entities (a total of seven relations that cover relations such as "part of" or "used for"), and (3) coreference links between the entities (indicating whether one entity refers to the other). An example of annotation conducted for the dataset is given in Figure 1. The authors trained a multitask model that detects all three annotation levels (*i.e.*, entities, relations, and coreference links) and used the trained model for extracting entities, relations, and coreference links from a corpus of scientific abstracts. The model is a neural network containing a number of layers used for constructing *span embeddings* [39] for each span of words in the text (*i.e.*, one or more consecutive words). The entities and relations were extracted from a corpus of abstracts from 12 AI conference proceedings and then arranged in a knowledge graph used for the subsequent analysis of scientific trends in AI publications.

In a more recent work by Jain *et al.* [30], a comprehensively annotated dataset (SciREX) of document-level relations in scientific articles

was introduced with annotations of four entity types: datasets, metrics, tasks, and methods. In the dataset, all entities were extracted, and their mentions, together with coreference links and document-level relations between entities, were annotated. SciREX contains a total of 438 annotated articles, and the authors emphasize a large number of relations that span across sentences, which justifies the need for document-level relation extraction, as opposed to most of the previous work. An example of entity-mentioning annotations in SciREX is shown in Figure 2. The authors also present a strong neural-based baseline model that takes an article's text as input and outputs extracted entities and relations among them.

Systems capable of extracting the key aspects or entities from scientific articles make it possible to transform the raw text of scientific articles into more structured representations, which can then be used for other tasks that use scientific articles as input. These representations can then be fed as input to systems dealing with various downstream tasks of scientific text processing. For example, the structure presented in Figure 1, showing the relations between entities found in the article's text, could be used to generate an explanation as to why this article is recommended for citation.

3. Citation Function Classification

Scientists cite articles for different reasons: sometimes it is because they use the methods described in the cited article, other times it is because they are attempting to solve the same problem as in the cited article. Based on different reasons for citing, which have become commonly referred to as *citation function* (CF), a number of citation classification schemes have been proposed in the literature [1, 12, 33, 56, 62]. While the proposed schemes differ with respect to the number and granularity of CF classes, most of them agree on three basic classes:

1. *background* (citing an article with a similar background or one addressing a similar problem),
2. *method* (citing an article that describes the method used), and
3. *comparison* (comparing the results with those of the cited article).

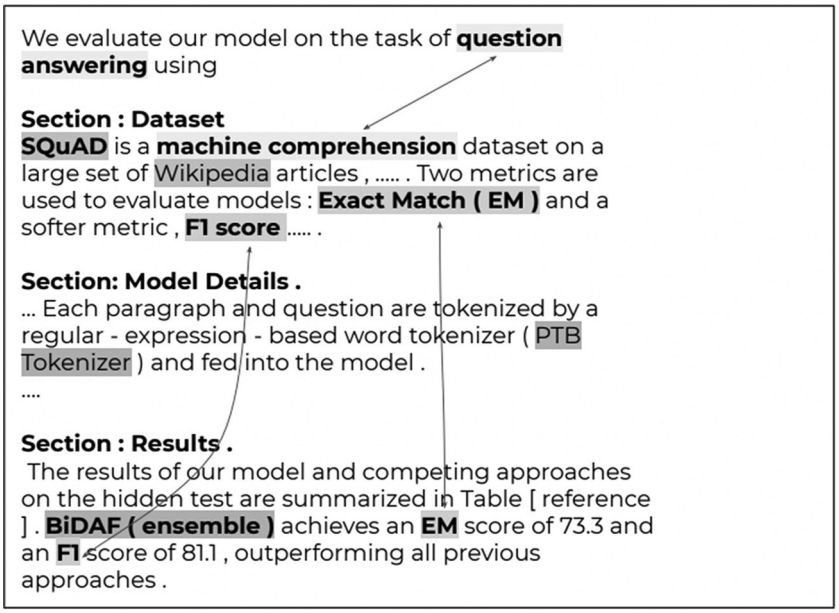

Figure 2. An example of annotation from the SciREX dataset. Entities, metrics, tasks, and methods are displayed in differently shaded background colors. Arrows indicate coreference links between different mentions. (Reprinted from [30] under Creative Commons-BY-4.0 license. Original figure was printed in color.)

This section outlines some of the most prominent work in CF classification, with a brief overview of models used for the task, based on both traditional machine learning and deep learning.

3.1. Traditional Machine Learning Approaches

Following the early work by Spiegel-Rosing [56], who was the first to propose a citation classification scheme, Teufel *et al.* [62] presented a CF annotation scheme consisting of 12 categories, grouped into 4 major categories:

1. explicit statement of weakness,
2. contrast or comparison with other work,
3. agreement/usage/compatibility with other work, and
4. neutral category.

The authors annotated a corpus of citation contexts from 360 articles in the field of computational linguistics and reported an inter-annotator agreement (kappa coefficient) of 0.72, which indicates high agreement. The distribution of categories in the annotated corpus is very skewed, with more than 60% of citations annotated with the neutral category. The annotated corpus is used for training a *k*-nearest neighbor classifier to predict the CF class for an input context. Contexts are represented with a variety of textual features, most relying on predefined sets of words that are indicative of the citation function. For example, a set of verbs conveying the meaning of *presentation* includes the following verbs: *propose*, *present*, *report*, and *suggest*. Some nontextual features they use include the relative position of the citation in the article, verb tenses, and verb modality. The authors report macro-F1 classification accuracy of 0.57 on a holdout test set.

Abu-Jbara *et al.* [1] proposed a different annotation scheme comprising only six categories: *criticizing*, *comparison*, *use*, *substantiating*, *basis*, and *neutral*. In addition to training a CF classifier, they also trained models for citation context identification and citation polarity classification (where polarity can be either positive, negative, or neutral). For citation context, they used the sentence in which citation occurs together with one sentence before and two sentences after it. The model uses a number of lexical and structural features, including the number of citations in the input context, a binary feature indicating whether the target citation appears in a group of citations or separately, and the verb/adjective/adverb that is the closest to the target citation. The authors evaluated their approach on a labeled subset of the ACL dataset and reported as the best result a macro-F1 score of 0.58 obtained using an SVM classifier with linear kernel. Distribution of the labels in the subset was quite skewed, with 47% of instances labeled as neutral.

Jurgens *et al.* [33] presented their own annotation scheme in which, unlike in the above described schemes, *comparison* and *contrast* are collapsed into a single class. Their scheme contains the following six classes: *background*, *motivation*, *uses*, *extension*, *comparison* or *contrast*, and *future*. The annotated dataset, extracted from articles sampled from the ACL dataset, totals 1,969 citation contexts in which the majority of contexts were annotated with background class. The authors proposed a model with features extracted from various structural, lexical, and metadata information from the input contexts. Similar to [1], they used predefined lists of verbs signalizing connective phrases, as well as function patterns presented in [60]. Metadata features used information about an article's venue, journal, number of citations per article and section, *etc.* The best results were achieved using a random forest classifier which, on their dataset, outperformed the previous state-of-the-art model of [1].

3.2. Deep Learning Approaches

Although the models described up to this point were all successful in the CF classification task, they all involved a handful of manually designed features extracted from both textual and metadata information. Following advances in deep learning (DL), a subfield of machine learning based on artificial neural networks, various DL-based mechanisms for NLP have become standard tools for developing DL models. Here, we briefly review some of them that were also used in prominent DL work on CF classification.

Standard deep learning modules for NLP. A standard method for representing words as input to DL-based NLP models is via *word embeddings*. Word embeddings are vector representations of words that enforce that words that often appear together – and hence are by virtue of the distributional hypothesis [63] also semantically related – have similar vector representations. The most popular algorithms for obtaining such representations are word2vec [44] and GloVe [48]. A standard building block for sequence processing in NLP via DL models is a long short-term memory (LSTM) cell [27], a type of recurrent neural network [53] that uses information from words previously seen in a sequence to produce a representation of the current word. The obtained word representations are *contextualized* in that they capture the context surrounding the word in a sequence, as opposed to representations obtained via word2vec or GloVe, which always produce the same representation for a word, regardless of its context. Since a single LSTM cell can only be applied in one direction, a bidirectional LSTM cell [55] is typically used to capture both sides of the context.

An LSTM cell outputs a single representation for each individual word in a sequence. To obtain a representation of the entire sequence, word representations of the individual words are typically averaged into a single vector representation. Better sequence representations can be obtained by the use of *attention mechanisms* [3], which enable soft pooling over word embeddings, yielding a weighted average over word representations and accordingly allowing the models to focus more on some parts of a sequence. The most recent trend, however, is the use of the transformer architecture [65], which leverages several attention mechanisms to generate contextual representations and eliminates recurrence in favor of attention-based feedforward processing. Building on the transformer architecture, Devlin *et al.* [15] introduced BERT, a model consisting of a number of transformer layers, which readily attained state-of-the-art results on various NLP tasks. The main idea behind BERT is to pretrain a language model using the task of predicting a missing token in the input sentence – a task that has been shown to induce some level of general linguistic competence [59]. Token and sentence representations obtained using BERT can then be transferred to other NLP models, *i.e.*, used as inputs to models for various specific NLP tasks.

Deep learning models for citation function classification. Following the advances in DL, which led to improvements in various NLP tasks, new state-of-the-art models were proposed by Cohan *et al.* [12] and Beltagy *et al.* [4]. Both approaches rely on deep learning architectures for training CF classifiers, and more importantly, they outperform all previously proposed models while relying only on textual information.

Cohan *et al.* [12] adopted multitask learning [9, 58] to train a model for citation function classification. This popular machine learning paradigm trains a machine learning model for a specific task by training the model jointly for that task and a number of additional tasks. This allows the model to recognize similarities across tasks and use this to generalize better than when training the models separately for each. Cohan *et al.* [12] trained a multitask model for learning a shared representation of citation context used in three tasks:

1. CF classification,
2. predicting the title of a section in which a citation occurred, and
3. predicting the worthiness of a citation (*i.e.*, whether current input warrants a citation).

An overview of the proposed model is given in Figure 3. The model uses pretrained word embeddings obtained with GloVe [48], and contextualized embeddings from ELMo [49] as input. GloVe embeddings are produced using an unsupervised algorithm that uses aggregated global word co-occurrence statistics and produces a single word embedding for each word in the corpus. ELMo embeddings are outputs of a deep pretrained language model (*i.e.*, a model that takes a sequence of words as input and outputs the most likely next word) which consists of a number of layers and outputs the representation of words that captures the context in which words appear. The model trained in the multitask setup uses a bidirectional LSTM cell [55] to produce hidden states over all the steps in the input citation context, which are then used for constructing context representation. Additionally, they employ an attention mech-

anism over hidden states to construct the representation of context, which allows the model to focus on more informative parts of the input. Finally, multitask learning is accomplished by using a separate multilayer perceptron for each task, taking the attention-weighted context representation (z) as input to each perceptron. The model is then optimized to minimize the weighted sum of all three losses (L_1, L_2, L_3). The authors evaluated their approach on ACL and the newly released SciCite dataset, comprising 11,020 citation contexts annotated with three citation function categories: background, method, and result or comparison. In both datasets, the majority of citation contexts are annotated with background category. The comparison with the previous state-of-the-art model of [33] shows an improvement on both datasets, with ACL macro-F1 score improving by over 10% (from 54.6% to 67.9%).

In an effort to adapt BERT for scientific corpora, Beltagy *et al.* [4] released a pretrained language model (SciBERT), trained on a corpus of scientific articles, and evaluated representations obtained with the model on a variety of tasks in the domain of scientific articles, including CF classification. SciBERT follows the BERT pretraining [15], a language model based on the transformer architecture [65], which leverages several attention mechanisms to generate contextual representations of tokens in the input sentence and the representation of the sentence. Beltagy *et al.* [4] used the same pretraining approach as Devlin *et al.* [15], but instead of training the model on a general corpus, they used a corpus of 1.14 million scientific articles, thus obtaining representations more specific to the scientific domain. They transferred the pretrained SciBERT embeddings to the task of CF classification and evaluated a model using those embeddings on both ACL and SciCite datasets, showing improvements of 3% and 1.5% (respectively) in macro-F1 over the multitask approach by Cohan *et al.* [12].

Understanding reasons behind citing a certain article offers scientists a better overview of the relations between various articles in the field they are interested in. Such information can help them detect articles with high influence in different aspects of the field, such as the field's background or methods often used in the field. Apart from that, knowing the reason behind the citation can be used in other downstream, citation-related tasks, such as citation recommendation.

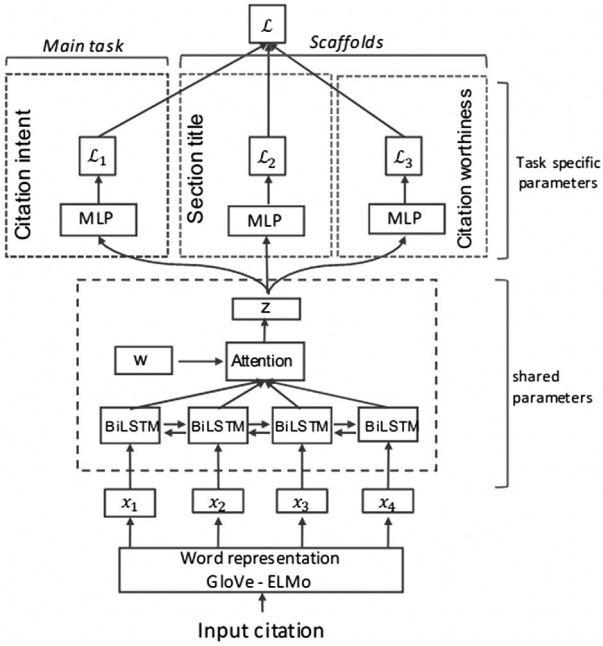

Figure 3. Overview of a multitask approach by Cohan *et al.* [12]. *Scaffolds* represent additional tasks that are used to help the model learn a better representation for the main task (*i.e.*, citation function classification). (Reprinted from [12] under Creative Commons-BY-4.0 license.)

4. Citation Recommendation

Recall from the introduction that citation recommendation (CR) is the task of recommending relevant articles that should be cited in a given article [8]. For example, given an article's manuscript, a CR system should return a list of relevant articles across all articles available in a scientific database. Citation recommendation systems may be divided into two groups based on what they use as input: *global CR systems* and *local CR systems*. Global CR systems consider a draft or a manuscript of an article and generate general ("global") recommendations for it, considering the entire text of the article as input, while local CR systems look at the part of the article where the citation is located, *i.e.*, the context of the citation, and suggest citations for that specific context. Both variants can be framed as a document retrieval problem, in which queries are either article drafts or citation contexts, and the documents to be retrieved are

scientific articles in the collection of articles. Figure 4 displays the difference between a typical global and local CR system.

This section describes the two variants of the task and gives an overview of the available datasets, evaluation metrics, and models used for the tasks. For a more detailed overview of the tasks and approaches, we refer the reader to the work of Färber and Jatowt [17].

Global citation recommendation. As mentioned, global CR systems take a draft of an entire article as input and produce a list of articles that should be cited in the article draft. Ideally, such a system is trained on project ideas as drafts, since that is typically when scientists search for potential references or starting points for their research. However, such a scenario is difficult to emulate, mainly because datasets of project ideas together with the relevant references are costly to acquire. As a proxy, in most cases, the article's abstract is used as a draft, and all the articles referenced in that article are considered those that should be cited.

Local citation recommendation. Although the recommendations obtained with global CR systems are helpful and serve as a good starting point for further literature surveys, they can sometimes be too broad for particular problems discussed in a scientific article. Local CR systems attempt to address this by using the text around the citation in the article, commonly referred to as *citation context*, as the query for finding relevant articles.

Typically, a citation context involves a sentence in which citation occurred, *i.e.*, the citing sentence, together with a number of sentences before and after the citing sentence. When using the citation context as input to a local CR system, the citation is masked with a placeholder and the system is tasked to predict which of the articles from the collection is cited in that particular context. Therefore, from a standard document retrieval perspective, a citation context corresponds to a query, while the article corresponds to a document. As an example, consider the following fragment from an article, featuring a citation[6]:

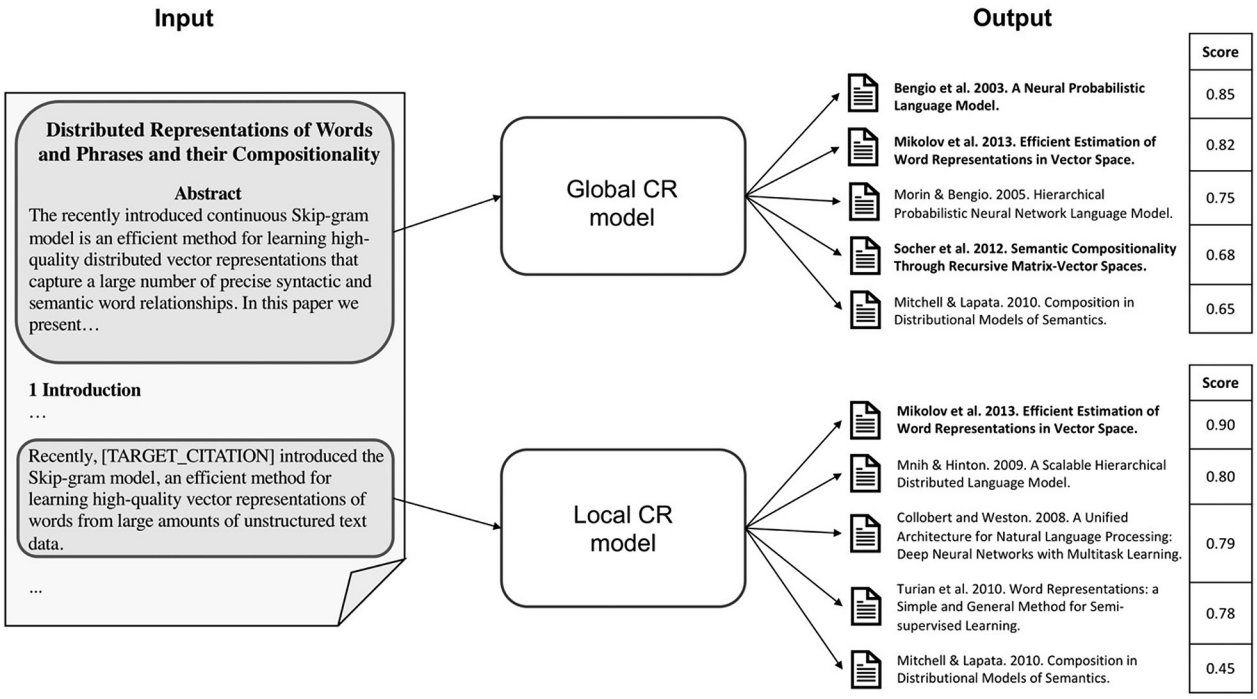

Figure 4. Difference between a global and a local CR system. The upper part shows a typical input to a global CR model (the title and abstract of the citing article), while the lower part shows a typical input to a local CR system (a citation context where citation is masked with a placeholder "TARGET_CITATION"). In both cases, the output contains a ranked list of articles ordered by relevance scores. Articles with names printed in bold represent correct recommendations.

We obtain the word embedding by training an unsupervised word2vec [CITATION] model on the training and validation splits and then use the word embedding to initialize W_e.

A successful local CR system should recommend the correct citation for the "[CITATION]" placeholder in the given example. In this case, it is the article "*Distributed representations of words and phrases and their compositionality*" by Mikolov *et al.* (2013). Note that, in general, and unlike in this case, there can in principle be a number of correct citations for a given citation context, for instance, in cases where there is a set of equally relevant articles that can be cited.

4.1. Datasets

To train and evaluate CR models, a variety of datasets of scientific articles have been compiled to this date, most of which are for the English language (a few datasets for languages other than English exist, *e.g.*, Chinese [32]). Datasets used for CR typically comprise two components: a set of scientific articles (each with its textual and metadata information) and accompanying information about citations across articles. The latter effectively defines a *citation graph*.

Datasets differ with respect to the research fields they cover, as well as the data available for articles and citations. For example, some datasets contain full article texts, while others contain only the title and abstract of each article. The reason why some datasets do not contain full texts or articles has mostly to do with copyright restrictions, as many articles are not in open access. A further point of difference between datasets is in the amount of citation information. While some datasets list only the articles cited in an article, others provide textual contexts for those citations as well. The consequence of these differences is that some datasets are suitable only for a global CR task (datasets that do not contain contexts), while others can be used for both global and local CR tasks. Below, we give a brief description of the commonly used CR datasets.

ACL-ARC.[7] The ACL-ARC dataset (*Association for Computational Linguistics – Anthology Reference Corpus*) comprises scientific articles compiled from conferences and journals from the field of computational linguistics and natural language processing. All the articles are from publications (journals or conference proceedings) published by the ACL (*Association for Computational Linguistics*[8]), an international scientific society for researchers working in the field of computational linguistics and natural language processing. Initially, released in 2008 by Bird *et al.* [7], the dataset has since been updated a number of times. While all dataset versions contain full texts of articles, authors, and venue information, as well as a citation graph across all the articles, the version from 2016 additionally contains automatically extracted citation contexts for 22,878 articles. In this version, citation contexts are extracted from PDF files using ParsCit [14] and contain 600 characters before and after a citation marker in the context, which often spans several sentences. The inclusion of citation contexts makes this dataset suitable for the local CR task.

RefSeer. The RefSeer dataset [28] was obtained from the collection of articles curated by the CiteSeer digital library[9] [18]. Each article is represented with the title, abstract, author, and venue information. The dataset contains over 800 000 articles from various domains, the majority of which are from the computer science domain, and over 4.5 million citation pairs over these articles. All citations are provided with the corresponding citation context, where each citation context contains 200 characters before and after the citation marker.

DBLP.[10] Introduced in [68], the DBLP dataset contains articles from the computer science domain and provides information about the ti-

[6]The excerpt is taken from the article: "Hierarchical Attention Networks for Document Classification" by Yang *et al.* (2016).

[7]https://acl-arc.comp.nus.edu.sg

[8]https://www.aclweb.org

[9]https://citeseerx.ist.psu.edu

[10]https://dblp.dagstuhl.de

tle, abstract, author, and venue information for each article. Although the database is regularly populated with new articles, the dataset version used in most papers on CR contains over 50 000 articles with an average of five citations per article.

PubMed.[11] The PubMed dataset contains articles from the field of biomedicine. The version used in most CR papers contains over 45 000 articles with an average of 17 citations per article [6]. Articles are represented with their title, abstract, author, and venue information.

OpenCorpus.[12] The Open Corpus dataset contains approximately 7 million articles, mostly from computer science and neuroscience domains [6]. Articles in the dataset contain title, abstract, author, year, venue, keyphrases, and citation information. There are no citation contexts, only the information about citing and cited articles.

S2ORC.[13] The S2OCR (The Semantic Scholar Open Research Corpus) dataset by Lo *et al.* [41] is the most recent dataset of scientific articles and accompanying metadata. The dataset contains entries for 81.1 million academic publications from various scientific domains, including full article texts for 8.1 million freely accessible publications together with detected citations, tables, and figures. In addition to article texts, the dataset also contains citation contexts linked to corresponding entries in the dataset.

4.2. Evaluation

Evaluation of citation recommendation systems, both in global and local setups, is performed using standard metrics adopted from recommendation system evaluation. These metrics evaluate system-generated lists of recommended items, where items are sorted by relevance. The point of difference between the evaluation of global and local CR systems is primarily in how many items are being considered as correct recommendations. In a typical global CR setup, there are many correct items, as one article cites many other articles. In contrast, in a local CR setup, often only one article is considered a correct recommendation, based on the fact that the article's author or authors cited only one article in the given context. As already mentioned, this does not, in principle, mean that citation context does not warrant other citations. Rather, because the local CR datasets are derived from existing articles and citations, they are by design limited to one citation per context.

A deeper conceptual problem with current CR evaluation setups is that the citation data (cited articles) are assumed to be the ground truth. In reality, however, those citations are extracted from existing citations made by article authors, who themselves used some method for finding potential references and then in many cases chose just a few to cite in their work. It may, therefore, well be the case that in many citation contexts citing articles other than the ones actually cited may be equally well or even better justified. In some cases, this can even lead to certain biases, *e.g.*, favoring the citation of less suitable but high-profile articles over those that are more suitable to be cited in a given context but are less popular. To remedy this, ideally, CR datasets used for both evaluation as well as training would be constructed over the complete set of available articles by asking human judges to rank the articles according to their relevance for the given input (either a manuscript or a citation context). While compiling datasets in this way is hardly feasible due to the high human labor cost, it is worth considering whether a more cost-efficient procedure could be set up to at least improve dataset completeness. Additionally, while evaluation is performed on existing CR datasets, one should keep this inherent limitation in mind, especially when gauging the precision of a CR system.

CR systems are usually evaluated using time-based data splits, *i.e.*, training, validation, and test sets are constructed based on the years in which the citing articles were published. Such splits offer a more realistic evaluation compared to standard random data splits since, in reality, a CR system can only recommend articles that were published before the citing article.

[11] https://www.ncbi.nlm.nih.gov/pubmed
[12] https://api.semanticscholar.org/corpus
[13] https://github.com/allenai/s2orc

Evaluation of a CR system amounts to comparing the output of the system for each article (global CR) or citation context (local CR) in the test set to reference citations in the dataset (serving as ground truth). Based on this comparison, system performance may be quantified using different metrics, which we discuss below.

Precision, recall, and *F1* at *k*. Precision and recall at *k*, denoted *P@k* and *R@k*, respectively, are calculated over the top *k* recommended items in a ranked list of recommended items, as follows:

$$P@k = \frac{TP}{k}, \quad R@k = \frac{TP + FN}{k}, \qquad (1)$$

where *TP* is the number of relevant items in the top *k* recommendations (*i.e.*, *true positives*) and *FN* is the number of relevant items that are not included in the top *k* recommendations (*i.e.*, *false negatives*). In general, precision at *k* captures the percentage of correct recommendations in the top *k* recommended items, while recall at *k* captures the percentage of overall correct recommendations that are included in the top *k* recommendations. The *F1@k* is calculated as the harmonic mean of the corresponding *P@k* and *R@k* scores and is used to obtain a single metric that balances precision and recall, therefore providing a better overview of the system's performance. All three metrics take values between 0 and 1, with higher values indicating a better result.

Mean reciprocal rank. Although precision, recall, and *F1@k* provide a useful measure of model performance, these metrics are rank-insensitive, *i.e.*, treat equally all correctly recommended items regardless of whether they are ranked first or *k*-th. Reciprocal rank is a statistical measure commonly used in information retrieval that calculates the rank of the first correct item in the ranked list of recommended items. When multiple queries are used in the evaluation, the mean reciprocal rank (MRR) is calculated as the average of reciprocal ranks across all queries, as follows:

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

where *Q* is a set of queries and $rank_i$ is the rank of the first relevant item in the list of items for the *i*-th query. As in the case of precision, recall, and *F1@k*, MRR also takes values between 0 and 1, with higher values indicating a better result. MRR is useful for situations in which ranking a single relevant item higher is more important than ranking other relevant items higher as well, since the metric takes into account the rank of only the first relevant item.

Mean average precision. Average precision is calculated taking into account all relevant items in a ranked list of recommended items:

$$AvgP(q) = \frac{\sum_{k=1}^{n}\left(P@k \cdot rel(k)\right)}{Rel_q}$$

where *n* is the number of recommended items, *P@k* is precision at *k* as defined by (1), *rel*(*k*) is an indicator function that equals 1 if the item at rank *k* is a relevant item and zero otherwise, and Rel_q is the total number of relevant items for the input query *q*. Mean average precision (MAP) is calculated as the average of average precisions over a set of queries:

$$MAP = \frac{\sum_{q=1}^{|Q|} AvgP(q)}{|Q|}$$

MAP also takes values between 0 and 1, with higher values meaning better performance. MAP metric is useful in situations when ranking all relevant items high is important, as opposed to ranking a single item higher than others, as is the case with MRR.

Normalized discounted cumulative gain. Generally, not all items need to be equally relevant for a query. When, given a query, a number of items may be suitably recommended, but they differ in the degree of relevance, then neither of the above described measures can adequately compare the performances of the models that produce different recommendation rankings. To compare the model's recommendations featuring graded relevance scores, discounted cumulative gain (DCG) [31] is typically used. DCG is a measure of ranking quality that measures the gain of recommended items based on their position in the ranked list, in which the items are sorted by their graded relevance scores. Normalized DCG is calculated in three steps: non-normalized DCG (DCG), followed by ideal DCG (IDCG), and then normal-

ized DCG (NDCG). The non-normalized DCG is calculated as follows:

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)} \quad (2)$$

where p is the rank position for which the metric is calculated and rel_i is the graded relevance of item i (in the case of binary relevance, it is either zero or one). Relevance scores are inversely scaled with the logarithm of the rank, which has an effect of penalizing relevant items that appear lower in the ranked list. Since the number of relevant items in the ranked lists differs across queries, expression (2) is normalized with ideal *DCG* (denoted *IDCG*) for each query in the set of queries.

The IDCG is defined as follows:

$$IDCG_p = \sum_{i=1}^{|REL_p|} \frac{rel_i}{\log_2(i+1)}$$

where REL_p is the list of relevant items ordered by their relevance up to position p. This normalization treats queries differently depending on the number of relevant items for each query. Normalized discounted cumulative gain (NDCG) for single query q at position p is then computed as:

$$NDCG_p = \frac{DCG_p}{IDCG_p} \quad (3)$$

Averaged values obtained by (3) across all the queries give the final measure for a set of queries, which takes values from 0 to 1 and a higher value means a better result. Similar to MAP, NDCG is also useful for situations in which ranking all relevant items highly is important. However, compared to MAP, NDCG penalizes lower placement of relevant items more, since it inversely scales item's relevance score with the logarithm of the rank.

Both global and local CR datasets are typically constructed using parsers that parse the texts of citing articles and provide a structured output containing all the references (*i.e.*, cited articles) and citation contexts in which each reference is cited. While such output is appropriate for global CR, it is not suitable for local CR approaches, as citation contexts in which more than one article is cited do not contain all the cited articles as true positive citations but only a single cited article per context. For this reason, researchers often use various string matching techniques to match citation contexts in which several articles were cited together to obtain a more realistic dataset for both training and evaluation. However, most of the work on local CR disregards this dataset deficiency and uses datasets with only one cited article per context being considered true positive.

4.3. Models for Global Citation Recommendation

Early work on global CR focused on keyword search across a pool of articles combined with filtering based on citation counts and author networks [29, 36, 43]. Those systems used either the full texts of articles or a few keywords as queries.

Bethard and Jurafsky [5] devised a more realistic approach in which they used the title and abstract of a scientific article as a proxy for a research idea and treated all the referenced articles as recommendations. Although using only the title and abstract as the query is more realistic than using the full text of the article, they admit that this is still far from ideal, as abstracts typically contain information about the results of the conducted research, which is something research ideas do not have. Furthermore, by looking only at the title and abstract of the article, one could hardly list all the references that appear in that article, since the abstract by definition offers a very condensed description of the article's content. Using such a proxy setup, with abstract and title as queries, the authors proposed a CR model based on both classical textual and metadata features. The textual features used include (1) TF-IDF scores of terms appearing in the title and abstract of query articles, (2) terms appearing in the texts of possibly referenced articles (documents), and (3) topical embeddings over articles. Metadata features include article and author citation counts, article PageRank scores [46], the number of years since the paper was published, and others. The authors also collected all the citation contexts in which an article was cited and vectorize these contexts using TF-IDF. The final recommenda-

tion score is calculated as a weighted sum of all the feature scores, where feature weights are learned through an iterative process. In each iteration, an underlying model (logistic regression or SVM) is used both for optimizing the weights of the recommendation model and for dataset expansion with additional article candidates. In the expansion phase of the iteration, candidate articles that are recommended by the current version of the model are labeled as either correct or incorrect recommendations, based on the reference list that is available for each query article. These recommendations are then added together with their labels to the training set to be used in the next iteration. The motivation for this procedure is to allow the model to gradually improve the performance, as opposed to training a model only once on a large training set. The proposed models are evaluated on the ACL-ARC dataset using time-based splits, which, as mentioned at the beginning of this subsection, offer a more realistic evaluation scenario as opposed to random splits. In the evaluation, SVM with linear kernel is reported as the best performing model, reaching a MAP score of 28.7 on the test set.

The work by Bethard and Jurafsky [5] demonstrated that CR models can perform well on single-domain datasets (the ACL-ARC dataset). In contrast, Ren *et al.* [52] addressed the problem of cross-domain CR. They adopted a graph-based approach to CR, emphasizing the need for different citation treatments based on the author's information need (or *intent*) behind the citation. The proposed system, named ClusCite, uses soft clusters of articles based on so-called *interest groups*, which are constructed over a set of articles, authors, and venues. When generating a recommendation for a given input manuscript, ClusCite is tasked with recommending articles that are more relevant to the intent described in the manuscript, by focusing on the intent's *interest groups* and the articles with high ranking scores in those groups. The scoring function for recommendation thus uses two cluster-based functions: one that measures the relatedness between input query q and each interest group k, and one that computes the relative importance of paper p within interest group k. Together with textual features, ClusCite uses a variety of graph-extracted features to represent articles and venues and constructs corresponding interest groups. The model was evaluated on the DBLP and PubMed datasets, with MRR reaching over 0.5 on both datasets. However, the good performance comes at a cost of complexity, as the time complexity of training the model increases with the number of links in the dataset.

Advances in deep learning have motivated novel, DL-based approaches to global CR. In DL, the similarity between items using DL is typically modeled by first constructing a vector representation for each item (an *embedding*) and then using a metric in the embedding space (*e.g.*, cosine similarity or L2-distance) to compute the similarity. In a work motivated by representing articles in shared embedding space, Bhagavatula *et al.* [6] presented a content-based method for citation recommendation that is robust to the lack of metadata in queries (*e.g.*, missing author names or initial list of citations). The presented system comprises two modules: candidate selection and reranker. Figure 5 gives an overview of both modules. In the candidate selection phase, all articles from the dataset are projected into the same embedding space using a neural network module that learns how to construct the article's embedding. This neural network uses words from the article's title and abstract represented via their word embeddings. The NNSelect module is trained via triplet-based metric learning. Triplet-based metric learning [54] works by comparing three items (triplets), one of which is an anchor (a reference item), another a positive item (an item considered similar to the anchor), and yet another a negative item (an item considered not similar to the anchor). The metric calculates the similarity between the anchor and the positive item, as well as between the anchor and the negative item. In a DL-based model, similarity is calculated via a neural network module, and the proposed metric can thus be used for calculating the loss of the network: if the similarity between negative item and the anchor is higher than the similarity between the anchor and the positive item, the loss will be high and the network will be penalized. The training of NNSelect is carried out on a training set made of triplets (d_q, d^+, d^-), where d_q is the query article, d^+ is an article cited in d_q (positive item), and d^- is an article not cited in d_q (negative item). Nega-

tive items used in the training set of triplets are selected in three ways:

1. at random,
2. by selecting the nearest neighbors from the embedding space that were not cited in d_q, or
3. by selecting the articles cited in the citations of d_q but not cited in d_q.

The second step continues with the top k closest neighbors of query articles from the article embedding space and uses a scoring model for deciding on the final recommendation score for neighboring articles. The scoring model (NNRank) is a small neural network that takes cosine similarities between different article fields (*e.g.*, title, abstract, authors) concatenated with the weights for words appearing in both articles, cosine similarity between the two documents from step 1, and the number of citations for documents that are being scored. The authors evaluate their approach on PubMed, DBLP, and OpenCorpus dataset, using *F1@k* and MRR measures. The results show significant improvement over previous state-of-the-art results by Ren *et al.* [52]. More specifically, the proposed approach outperforms ClusCite by 15% in absolute improvement on the DBLP dataset, and 19% on the PubMed dataset.

The results of Bhagavatula *et al.* [6] demonstrate that having good semantic representations (vector embeddings) is beneficial for CR. Along these lines, Cohan *et al.* [13] addressed the task of creating generic representations of semantic articles that could be suitable for CR, but also more generally for other scientific text processing tasks. The proposed model, named SPECTER, builds on SciBERT [4] and produces article embeddings based solely on the article's title and abstract. This is accomplished by fine-tuning SciBERT (*cf.* Subsection 3.2) on a dataset of scientific articles and

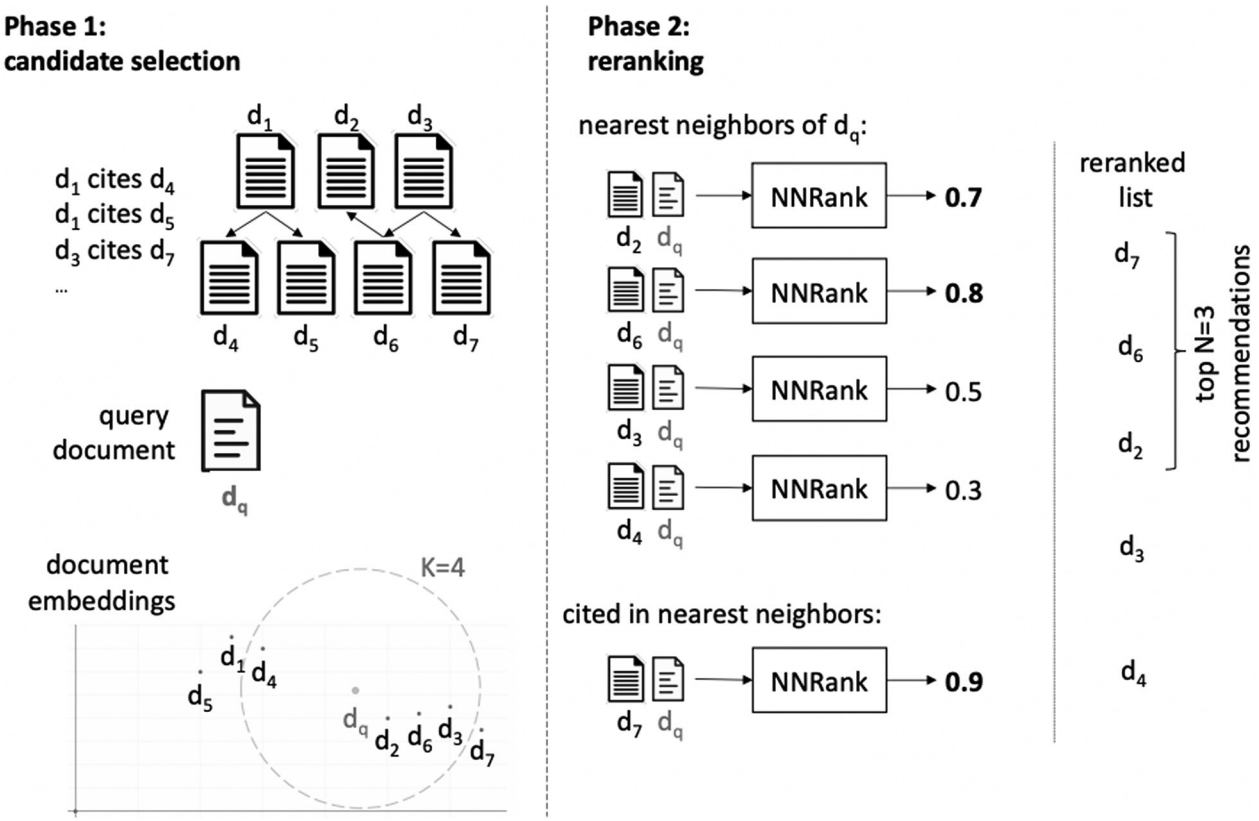

Figure 5. Overview of the global CR system presented in [6]. In Phase 1, all seven documents d are projected into a shared document embedding space, and K nearest neighbors for a query document are passed to Phase 2. In Phase 2, K candidate documents are reranked using the reranker model, and documents are then scored according to the obtained recommendation scores. (Reprinted from [6] under Creative Commons-BY-4.0 license.)

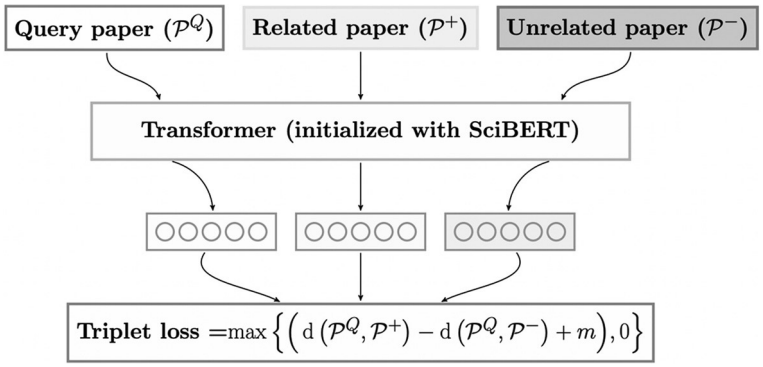

Figure 6. Overview of SPECTER [13]. All three input papers (query, related, and unrelated) are passed through the same transformer block to obtain paper embedding. Loss is calculated as triplet loss using the L2-norm between obtained paper embeddings. (Reprinted from [13] under Creative Commons-BY-4.0 license.)

citation links between them obtained from the Semantic Scholar database. Contrary to previous approaches for producing semantic representations of articles based solely on text, the proposed model uses information about citations between articles to minimize the L2-norm between embeddings of those articles that cite each other and maximize the L2-norm between those that do not, thus enforcing similar representations of articles that are in cited relation and therefore similar in some respect. An overview of the SPECTER model, together with inputs, is given in Figure 6.

As their evaluation shows, such embeddings can be used in a range of tasks without any task specific finetuning of the presented model. Although the authors do not evaluate the proposed model on any specific global citation recommendation dataset, but rather on "clickthrough" data from a public scholarly search engine, they compare their model with the one proposed by Bhagavatula *et al.* [6] and obtain better results, *i.e.*, a 1.4% improvement in NDCG and 2.7% in *P*@1.

4.4. Models for Local Citation Recommendation

Recall from the beginning of this section that local CR is defined as the task of recommending articles for citing in a given context. Here, we briefly review some of the prominent work on local CR.

He *et al.* [24] were the first to introduce the problem of local citation recommendation. They compiled a corpus of over 400 000 articles from the CiteSeer system to be used as the training set for their model. Each article in the dataset is complemented with information about the authors, title, abstract, and all citation contexts that referenced that article. Compared to RefSeer, a dataset also derived from CiteSeer (*cf.* Subsection 4.1), the dataset by He *et al.* is restricted to the articles published before 2008, while those published in 2008 are used as test set articles, which resulted in the dataset being smaller than RefSeer. When predicting the missing reference in the input citation context, they calculate the similarity between the input context (together with its title and abstract) and candidate articles from the corpus, which are represented using TF-IDF scores of text from the title, abstract, and the corresponding citation contexts. Although the proposed approach performed well on the test set, the modeling of articles and contexts is based on the presence of certain words in articles, which is often not enough to represent the semantic meaning of longer phrases. Moreover, since the system uses article citations to decide whether it is relevant for the querying context, the approach is not suitable for recommending articles that have not been cited previously.

Following the advances in deep learning and their applications to NLP, Huang *et al.* [28] introduced a neural probabilistic model for local citation recommendation. The authors jointly train a model for learning the representation

of words in the context and the cited article. The joint model is trained by maximizing the dot product between the embeddings of two words from the same context and between the embedding of the cited article and citing context words. Forcing the embeddings of words that appear in the same context to be close to each other enables the model to group semantically similar words together, which leads to better generalization of the model. At test time, articles are sorted by the sums of the probabilities of input words citing the given candidate article, where probabilities are calculated as dot products of word and article embeddings, scaled using the sigmoid function. The authors evaluate the model on the RefSeer dataset (*cf.* Subsection 4.1) reaching an MRR score of 0.184 on the top 10 recommended articles. In forcing vectors of words co-occurring in citation context to be similar, the model assumes independence between words in context, which prevents it from modeling compositionality between words, *i.e.*, representing the meaning of a sequence as a function of the meanings of parts of the sequence, with respect to the manner in which these parts are combined [47]. Another deficiency of the model is the closed set of article embeddings that are learned in training, since only those embeddings can be recommended at test time.

Addressing the model's inability to model compositionality between words in the context, Ebesu and Fang [16] introduced an encoder-decoder-based model for predicting the title of the article cited in a given context. An encoder-decoder neural network architecture consists of two blocks: (1) an encoder, a block that constructs an embedding for the given input, and (2) a decoder, a block that takes the embedding produced by the encoder and uses it to produce the final output of the structure. In NLP, this type of architecture is commonly used for sequence-to-sequence models [57], which take a sequence of words as input and produce another sequence of words as output (*e.g.*, in a different language). The encoder uses a time delay neural network (TDNN) [67] to create a citation context embedding, which is then combined with author embeddings (representations of both citing and cited authors) to create the final context embedding to be used as input to the decoder. The task of the decoder is to reconstruct from this input the title of the cited article. A model with such an encoder-decoder architecture can use both the information about words appearing in the citation context and information about authors of citing and cited articles to construct context embeddings as input to the decoder. The decoder should then use this information to output the title that is the most suitable, and the assumption is that this information embedded together should be useful in the process (*e.g.*, information about the authors might reveal what type of work they usually cite). The model was evaluated on the RefSeer dataset (*cf.* Subsection 4.1) and achieved an MRR score of 0.267 on the top 10 recommendations.

Despite the improvement in modeling semantics of the context via TDNN, the model of Ebesu and Fang [16] relies only on the title of the cited article, which can hardly contain all the information relevant for citing an article. Incorporating both the information from the citation context and the abstract or full text of the cited article, Yang *et al.* [70] introduced a model based on stacked denoising autoencoders [66] for producing the embedding of cited articles (in combination with learned author embeddings) and bidirectional LSTM cells for embedding the citation context. The generated embeddings are then concatenated into a single embedding, which is passed through a neural network that decides whether the article should be cited in a given context. The model has attention mechanisms built into both citation and article embedding generation, allowing it to focus on more informative parts of both pieces of information. Evaluation is performed on the ACL-ARC, DBLP, and RefSeer datasets (*cf.* Subsection 4.1), with the MRR score on the RefSeer dataset reaching 0.277 on the top 10 recommendations. The presented model includes more information for the cited article than the approach in [16] through text from the abstract, or even full article, but represents them through bag-of-words vectors of words appearing in the text, as the autoencoder is trained to predict bag-of-words vectors for articles, not a sequence of words as they appear in the text. Just as the approach of Huang *et al.* [28], this kind of bag-of-words representation is oblivious to the compositionality of words in the sequence. A semantically richer representation of an article, for example, one that would enclose

the complete text from either the abstract or full article and that would also model the compositionality of words in the sequence, would probably lead to better results in recommendation. However, devising such a representation is still difficult, given the length of the input articles and the inability of standard neural network building blocks (*e.g.*, LSTMs) to capture the semantics of longer sequences.

5. Conclusion

Scientific articles are the primary means for disseminating research findings and knowledge in today's scientific and increasingly technical society. As the number of scientific articles is growing to levels that make it difficult even for scientists to keep track of recent research in their field, technology has stepped in to make search and access to scientific articles easier. Natural language processing and machine learning methods are now being increasingly used for automated analysis of scientific articles on a large scale. This article presented an overview of the main tasks and methods in this exciting domain, some based solely on the analysis of articles' textual content (detection of key aspects and entities) and others on the analysis of the citations between the articles together with the text (citation function and recommendation). Citation recommendation (CR), in particular, is a potentially high-impact task, poised not only for making access to scientific publications more efficient but also for directly improving the quality of scientific production. With this in mind, our overview focused on citation recommendations in both local and global setups, with an overview of the available datasets and metrics used for the evaluation of citation recommendation systems.

As our overview shows, citation recommendation and related tasks have attracted much research interest, and in fact, most systems and approaches we described have been proposed in the last couple of years. However, much still remains to be done for the wide adoption of CR systems. Future research on CR will likely have to address the following three main issues: (1) improving the performance of current CR models, (2) evaluating CR models on more realistic datasets, and (3) devising CR models capable of providing explainable recommendations.

Improving CR models. One potential path for improving the performance of CR models may be offered by the multitask learning paradigm [9], which would combine a number of tasks from the domain of scientific article analysis to profit from joint learning on multiple related tasks. A recent work by Khadka [34] demonstrated that using explicit features about citation functions inside a citation recommendation system leads to an improvement in performance. Similarly, in a setup resembling that of [12], one could attempt to learn a unique representation of citation context for both citation function classification and local citation recommendation, as citation function already offers the reasoning behind citing a specific article, which might help reduce the number of candidate articles to be cited in that context. In a similar way, key aspects or entities extracted from an article could be used in a recommendation model as a method for enhancing representations of both citing and recommended articles. Further improvements could perhaps be obtained through a combination of global and local approaches. As an example, a hybrid setup in which a system is tasked to provide a number of recommendations for a specific section of an article, instead of the article as a whole, might turn out to be more efficient than both global and local approaches because, in this case, the size of the context strikes a good balance between being too narrow (local approaches) and too wide (global approaches).

Realistic evaluation. As argued in Section 4, current evaluation setups of citation recommendation systems suffer from a bias towards specific articles, as training data are not obtained by annotation but rather contain citations extracted from published articles, which might be biased towards the author's own or other scientists' work. Constructing a dataset of manually annotated relevant articles for a given input, *e.g.*, citation context, would offer a more realistic and unbiased evaluation dataset for such systems. However, obtaining such a dataset is expensive, especially given the number of published scientific articles and domain knowledge needed for deciding whether an article is a good candidate for citing in a given citation context. An alternative to this costly approach would be

to use some form of crowdsourcing [51], which could provide less expensive annotations that would hopefully not be of much worse quality than those obtained from domain experts.

Explainable recommendations. There has been an increased awareness in the AI and machine learning communities that we need to focus our efforts on building a human understandable, explainable AI system [10, 20]. Explainable AI systems contribute to transparency, fairness, and safety [23] and in general facilitate the synergy between AI systems and human experts. With this in mind, an explainable citation recommendation system could be designed to provide reasons behind each recommended article, as this would help scientists gain a better understanding as to why each recommended article is relevant for input context or manuscript.

Scientific text analysis in general, and CR and related tasks in particular, hold promise for improving the way we do science. As the number of published research fields continues to grow on a daily basis and research in NLP continues to devise new methods, new research challenges and perspectives will undoubtedly arise. With this work, we hope to provide the first steps in understanding the research effort conducted so far in this interesting research field.

Acknowledgment

The first author has been supported by a grant from the Croatian Science Foundation (HRZZ-DOK-2018-09).

References

[1] A. Abu-Jbara *et al.*, "Purpose and Polarity of Citation: Towards NLP-based Bibliometrics", in *Proc. of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 596–606.

[2] W. Ammar *et al.*, "Construction of the Literature Graph in Semantic Scholar", in *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 3, 2018, pp. 84–91. https://doi.org/10.18653/v1/n18-3011

[3] D. Bahdanau *et al.*, "Neural Machine Translation by Jointly Learning to Align and Translate", in *Proc. of the 3rd International Conference on Learning Representations, ICLR*, 2015.

[4] I. Beltagy *et al.*, "SciBERT: A Pretrained Language Model for Scientific Text", in *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLPIJCNLP)*, 2019, pp. 3613–3618. https://doi.org/10.18653/v1/d19-1371

[5] S. Bethard and D. Jurafsky, "Who Should I Cite: Learning Literature Search Models from Citation Behavior", in *Proc. of the 19th ACM International Conference on Information and Knowledge Management*, 2010, pp. 609–618. https://doi.org/10.1145/1871437.1871517

[6] C. Bhagavatula *et al.*, "Content-Based Citation Recommendation", in *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2018, pp. 238–251. https://doi.org/10.18653/v1/n18-1022

[7] S. Bird *et al.*, "The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Researchin Computational Linguistics", in *Proc. of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, 2008.

[8] K. D. Bollacker *et al.*, "CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications", in *Proc. of the Second International Conference on Autonomous Agents*, 1998, pp. 116–123. https://doi.org/10.1145/280765.280786

[9] R. Caruana, "Multitask Learning", *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997. https://doi.org/10.1023/a:1007379606734

[10] S. Chakraborty *et al.*, "Interpretability of Deep Learning Models: a Survey of Results", in *Proc. of the 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 2017, pp. 1–6. https://doi.org/10.1109/uic-atc.2017.8397411

[11] J. Chen and H. Zhuge, "Summarization of Scientific Documents by Detecting Common Facts in Citations", *Future Generation Computer Systems*, vol. 32, pp. 246–252, 2014. https://doi.org/10.1016/j.future.2013.07.018

[12] A. Cohan *et al.*, "Structural Scaffolds for Citation Intent Classification in Scientific Publications", in *Proc. of the 2019 Conference of the North American Chapter of the Association for Compu-*

tational Linguistics: Human Language Technologies, vol. 1, 2019, pp. 3586–3596. https://doi.org/10.18653/v1/n19-1361

[13] A. Cohan *et al.*, "SPECTER: Document-level Representation Learning using Citation-informed Transformers", in *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2270–2282. https://doi.org/10.18653/v1/2020.acl-main.207

[14] I. G. Councill *et al.*, "Parscit: An Opensource CRF Reference String Parsing Package", in *Proc. of the LREC*, 2008, pp. 661–667.

[15] J. Devlin *et al.*, "BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding", in *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2019, pp. 4171–4186.

[16] T. Ebesu and Y. Fang, "Neural Citation Network for Context-Aware Citation Recommendation", in *Proc. of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1093–1096, 2017. https://doi.org/10.1145/3077136.3080730

[17] M. Färber and A. Jatowt, "Citation Recommendation: Approaches and Datasets", *International Journal on Digital Libraries*, vol. 21, no. 4, pp. 375–405, 2020. https://doi.org/10.1007/s00799-020-00288-2

[18] C. L. Giles *et al.*, "Citeseer: An Automatic Citation Indexing System", in *Proc. of the Third ACM Conference on Digital Libraries*, 1998, pp. 89–98. https://doi.org/10.1145/276675.276685

[19] N. Green, "Identifying Argumentation Schemes in Genetics Research Articles", in *Proc. of the 2nd Workshop on Argumentation Mining*, 2015, pp. 12–21. https://doi.org/10.3115/v1/w15-0502

[20] R. Guidotti *et al.*, "A Survey of Methods for Explaining Black Box Models", *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018. https://doi.org/10.1145/3236009

[21] Y. Guo *et al.*, "Identifying the Information Structure of Scientific Abstracts: An Investigation of Three Different Schemes", in *Proc. of the 2010 Workshop on Biomedical Natural Language Processing*, 2010, pp. 99–107.

[22] S. Gupta and C. D. Manning, "Analyzing the Dynamics of Research by Extracting Key Aspectsof Scientific Papers", in *Proc. of the 5th International Joint Conference on Natural Language Processing*, 2011, pp. 1–9.

[23] H. Hagras, "Toward Human-Understandable, Explainable AI", *Computer*, vol. 51, no. 9, pp. 28–36, 2018. https://doi.org/10.1109/mc.2018.3620965

[24] Q. He *et al.*, "Context-Aware Citation Recommendation", in *Proc. of the 19th International Conference on World Wide Web*, 2010, pp. 421–430. https://doi.org/10.1145/1772690.1772734

[25] K. Heffernan and S. Teufel, "Identifying Problems and Solutions in Scientific Text", *Scientometrics*, vol. 116, no. 2, pp. 1367–1382, 2018. https://doi.org/10.1007/s11192-018-2718-6

[26] K. Hirohata *et al.*, "Identifying Sections in Scientific Abstracts using Conditional Random Fields", in *Proc. of the Third International Joint Conference on Natural Language Processing*, 2008, pp. 381–388.

[27] S. Hochreiter and J. Schmidhuber "Long Short-Term Memory", *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. https://doi.org/10.1162/neco.1997.9.8.1735

[28] W. Huang *et al.*, "A Neural Probabilistic Model for Context Based Citation Recommendation", in *Proc. of the 29th AAAI Conference on Artificial Intelligence*, 2015.

[29] P. Ingwersen and B. Larsen, "Using Citations for Ranking in Digital Libraries", in *Proc. of the 6th ACM/IEEECS Joint Conference on Digital Libraries (JCDL'06)*, 2006, pp. 370–370. https://doi.org/10.1145/1141753.1141865

[30] S. Jain *et al.*, "SciREX: A Challenge Dataset for Document-Level Information Extraction", in *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7506–7516. https://doi.org/10.18653/v1/2020.acl-main.670

[31] K. Järvelin and J. Kekäläinen, "Cumulated Gain-based Evaluation of IR Techniques", *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002. https://doi.org/10.1145/582415.582418

[32] Z. Jiang *et al.*, "Cross-Language Citation Recommendation via Hierarchical Representation Learning on Heterogeneous Graph", in *Proc. of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 635–644. https://doi.org/10.1145/3209978.3210032

[33] D. Jurgens *et al.*, "Measuring the Evolution of a Scientific Field through Citation Frames", *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 391–406, 2018. https://doi.org/10.1162/tacl_a_00028

[34] A. Khadka, "Capturing and Exploiting Citation Knowledge for the Recommendation of Scientific Publications", PhD thesis, The Open University, 2020.

[35] S. N. Kim *et al.*, "Semeval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles", in *Proc. of the 5th International Workshop on Semantic Evaluation*, 2010, pp. 21–26.

[36] B. Larsen, "Exploiting Citation Overlaps for Information Retrieval: Generating a Boomerang Effect from the Network of Scientific Papers", *Scientometrics*, vol. 54, no. 2, pp. 155–178, 2002.

[37] A. Lauscher *et al.*, "An Argument-Annotated Corpus of Scientific Publications", in *Proc. of the 5th Workshop on Argument Mining*, 2018, pp. 40–46. https://doi.org/10.18653/v1/w18-5206

[38] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents", *International Conference on Machine Learning*, 2014, pp. 1188–1196.

[39] K. Lee *et al.*, "End-to-End Neural Coreference Resolution", in *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 188–197. https://doi.org/10.18653/v1/d17-1018

[40] M. Liakata *et al.*, "Corpora for the Conceptualisation and Zoning of Scientific Papers", in *Proc. of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*, 2010, pp. 2054–2061.

[41] K. Lo *et al.*, "S2ORC: The Semantic Scholar Open Research Corpus", in *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 4969–4983. https://doi.org/10.18653/v1/2020.acl-main.447

[42] Y. Luan *et al.*, "Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction", in *Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3219–3232. https://doi.org/10.18653/v1/d18-1360

[43] E. Meij and M. De Rijke, "Using Prior Information Derived from Citations in Literature Search", pp. 665–670, 2007.

[44] T. Mikolov *et al.*, "Distributed Representations of Words and Phrases and Their Compositionality", *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111–3119, 2013.

[45] H. Oldenburg, "Epistle Dedicatory", *Philosophical Transactions of the Royal Society*. https://doi.org/10.1098/rstl.1673.0001

[46] L. Page *et al.*, "The PageRank Citation Ranking: Bringing Order to the Web", techreport, 1999.

[47] F. J. Pelletier, "The Principle of Semantic Compositionality", *Topoi*, vol. 13, no. 1, pp. 11–24, 1994. https://doi.org/10.1007/bf00763644

[48] J. Pennington *et al.*, "GloVe: Global Vectors for Word Representation", in *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. https://doi.org/10.3115/v1/d14-1162

[49] M. Peters *et al.*, "Deep Contextualized Word Representations", in *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, pp. 2227–2237. https://doi.org/10.18653/v1/n18-1202

[50] V. Qazvinian and D. R. Radev, "Scientific Paper Summarization using Citation Summary Networks", in *Proc. of the 22nd International Conference on Computational Linguistics*, 2008, pp. 689–696. https://doi.org/10.3115/1599081.1599168

[51] A. J. Quinn and B. B. Bederson. "A Taxonomy of Distributed Human Computation".

[52] X. Ren *et al.*, "Cluscite: Effective Citation Recommendation by Information Network-based Clustering", in *Proc. of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 821–830. https://doi.org/10.1145/2623330.2623630

[53] D. E. Rumelhart *et al.*, "Learning Representations by Backpropagating Errors", *Nature*, vol. 323, pp. 533–536, 1986. https://doi.org/10.1038/323533a0

[54] M. Schultz and T. Joachims, "Learning a Distance Metric from Relative Comparisons", *Advances in Neural Information Processing Systems*, vol. 16, pp. 41–48, 2003.

[55] M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks", *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997. https://doi.org/10.1109/78.650093

[56] I. Spiegel-Rosing, "Science Studies: Bibliometric and Content Analysis", *Social Studies of Science*, vol. 7, no. 1, pp. 97–113, 1977. https://doi.org/10.1177/030631277700700111

[57] I. Sutskever *et al.*, "Sequence to Sequence Learning with Neural Networks", *Advances in Neural Information Processing Systems*, vol. 27, pp. 3104–3112, 2014.

[58] S. Swayamdipta *et al.*, "Syntactic Scaffolds for Semantic Structures", in *Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3772–3782. https://doi.org/10.18653/v1/d18-1412

[59] I. Tenney *et al.*, "BERT Rediscovers the Classical NLP Pipeline", in *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, 2019. https://doi.org/10.18653/v1/p19-1452

[60] S. Teufel, "Argumentative Zoning: Information Extraction from Scientific Text", PhD thesis, 1999.

[61] S. Teufel and M. Moens, "Summarizing Scientific Articles: Experiments with Relevance and Rhe-

torical Status", *Computational Linguistics*, vol. 28, no. 4, pp. 409–445, 2002. https://doi.org/10.1162/089120102762671936

[62] S. Teufel *et al.*, "Automatic Classification of Citation Function", in *Proc. of the 2006 Conference on Empirical Methods in Natural Language Processing*, 2006, pp. 103–110. https://doi.org/10.3115/1610075.1610091

[63] P. D. Turney and P. Pantel, "From Frequency to Meaning: Vector Space Models of Semantics", *Journal of Artificial Intelligence Research*, vol. 37, pp. 141–188, 2010. https://doi.org/10.1613/jair.2934

[64] R. Van Noorden, "Scientists May Be Reaching a Peak in Reading Habits", *Nature News*, 2014. https://doi.org/10.1038/nature.2014.14658

[65] A. Vaswani *et al.*, "Attention Is All You Need", *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

[66] P. Vincent *et al.*, "Extracting and Composing Robust Features with Denoising Autoencoders", in *Proc. of the 25th International Conference on Machine Learning*, 2008, pp. 1096–1103. https://doi.org/10.1145/1390156.1390294

[67] A. Waibel *et al.*, "Phoneme Recognition using Time-Delay Neural Networks", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, 1989. https://doi.org/10.1016/b978-0-08-051584-7.50037-1

[68] M. Weis *et al.*, "A Duplicate Detection Benchmark for XML (and Relational) Data", in *Proc. of the Workshop on Information Quality for Information Systems (IQIS)*, 2006.

[69] K. E. White *et al.*, "Science and Engineering Publication Output Trends: 2014 Shows Rise of Developing Country Output while Developed Countries Dominate Highly Cited Publications", *National Science Foundation; Social, Behavioral and Economic Sciences*, 2017.

[70] L. Yang *et al.*, "Attention-Based Personalized Encoder-Decoder Model for Local Citation Recommendation", *Computational Intelligence and Neuroscience*, pp. 1232581:1–1232581:7, 2019. https://doi.org/10.1155/2019/1232581

Received: January 2021
Revised: April 2021
Accepted: April 2021

Contact addresses:
Zoran Medić
Text Analysis and Knowledge Engineering Lab
Faculty of Electrical Engineering and Computing
University of Zagreb
Croatia
e-mail: zoran.medic@fer.hr

Jan Šnajder
Text Analysis and Knowledge Engineering Lab
Faculty of Electrical Engineering and Computing
University of Zagreb
Croatia
e-mail: jan.snajder@fer.hr

Zoran Medić was born in Metković, Croatia. He received his BSc in computing in 2014 and MSc degree in computer science from the University of Zagreb, Croatia, in 2016. He is currently pursuing PhD in computer science at the University of Zagreb and working as a research assistant in the Laboratory for Text Analysis and Knowledge Engineering, at the Department of Electronics, Microelectronics, Computer and Intelligent Systems, at the Faculty of Electrical Engineering and Computing, University of Zagreb. His research interests include scholarly document processing, citation analysis and deep learning for natural language processing.

Jan Šnajder was born in Zagreb, Croatia. He received his BSc degree in computing in 2001 and MSc and PhD degrees in computer science from the University of Zagreb, in 2006 and 2010, respectively. Since 2001 he has been a Researcher at the Department of Electronics, Microelectronics, Computer and Intelligent Systems at the Faculty of Electrical Engineering and Computing, University of Zagreb, where he currently holds the position of an associate professor. He is the author or coauthor of over 100 journal or conference papers. His research interests include natural language processing, with the focus on information extraction and text analysis for computational social science.