From Automatic Keyword Detection to Ontology-based Topic Modeling

Marc Beck*[1], Syed Tahseen Raza Rizvi*[1,2][0000−0002−4359−4772], Andreas Dengel[1,2][0000−0002−6100−8255] and Sheraz Ahmed[2][0000−0002−4239−6520]

[1] Technische Universität Kaiserslautern, Germany
{m_beck12, s_rizvi14}@cs.uni-kl.de
[2] German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
{Syed_Tahseen_Raza.Rizvi, Andreas.Dengel, Sheraz.Ahmed}@dfki.de

Abstract. In this paper, we propose a novel, two-staged system, for keyword detection and ontology-driven topic modeling. The first stage specializes in keyword detection in which we introduce a novel graph-based unsupervised approach called Collective Connectivity-Aware Node Weight *(CoCoNoW)* for detecting keywords from the scientific literature. *CoCoNoW* builds a connectivity aware graph from a given publication text and eventually assigns weight to the extracted keywords to sort them in order of relevance. The second stage specializes in topic modeling, where a domain ontology serves as an attention-map/context for topic modeling based on the detected keywords. The use of an ontology makes this approach independent of domain and language. *CoCoNoW* is extensively evaluated on three publicly available datasets **Hulth2003**, **NLM500** and **SemEval2010**. Analysis of results reveals that *CoCoNoW* consistently outperforms the state-of-the-art approaches on the respective datasets.

Keywords: Keyword Detection, Ontology, Topics, Topic Modeling

1 Introduction

Keywords are of significant importance as they carry and represent the essence of a text collection. Due to the sheer volume of the available textual data, there has been an increase in demand for reliable keyword detection systems which can automatically, effectively and efficiently detect the best representative words from a given text. Automatic keyword detection is a crucial task for various applications. Some of its renowned applications include information retrieval, text summarization, and topic detection. In a library environment with thousands or millions of literature artifacts, e.g. books, journals or conference proceedings, automatic keyword detection from each scientific artifact [28] can assist in automatic indexing of scientific literature for the purpose of compiling library catalogs.

* Equal contribution

In 2014, about 2.5 million scientific articles were published in journals across the world [39]. This increased to more than 3 million articles published in 2018 [15]. It is certainly impractical to manually link huge volumes of scientific publications with appropriate representative keywords. Therefore, a system is imminent which can automatically analyze and index scientific articles. There has been quite a lot of research on the topic of automated keyword detection, however most of the approaches deal with social media like tweets [4, 6, 7, 9, 10, 12, 13, 22, 26, 28, 32, 33].

A popular approach for keyword detection is representing text as an undirected graph $G = (N, E)$, where the nodes N in graph G correspond to the individual terms in the text and the edges E correspond to the relation between these terms. The most popular relation is term co-occurrence, i.e. an edge is added to the graph between nodes n_1 and n_2 if both corresponding terms co-occur within a given sliding window. The recommended window size depends on the selected approach and often lies in the range between 2 and 10 [19, 24, 34]. Duari and Bhatnagar [12] note that the window size w has a strong influence on the properties of the resulting graph. With the increase in w, the density also increases while the average path length between any two nodes decreases.

The assumption behind this sliding window is that the words appearing closer together have some potential relationship [34]. There are several variations of the sliding window, e.g. letting the window slide over individual sentences rather than the entire text and stopping at certain punctuation marks [19]. Duari and Bhatnagar [12] proposed a new concept named Connectivity Aware Graph (CAG): Instead of using a fixed window size, they use a dynamic window size that always spans two consecutive sentences. They argue that consecutive sentences are related to one another. This is closely related to the concept of *pragmatics* i.e. transmission of meaning depending on the context, which is extensively studied in linguistics [11, 16, 23]. In their experiments, they showed that the performance of approaches generally increases when they use CAGs instead of graphs built using traditional window sizes.

The first stage consists of a novel unsupervised keyword detection approach called Collective Connectivity-Aware Node Weight (CoCoNoW). Our proposed approach essentially combines the concepts of Collective Node Weight [6], Connectivity Aware Graphs (CAGs) [12] and Positional Weight [13] to identify, estimate and sort keywords based on their respective weights. We evaluated our approach on three different publicly available datasets containing scientific publications on various topics and with different lengths. The results show that CoCoNoW outperforms other state-of-the-art keyword detection approaches consistently across all three data sets. In the second stage, detected keywords are used in combination with the Computer Science Ontology CSO 3.1[3] [35] to identify topics for individual publications.

The contributions of this paper are as follows:

[3] https://cso.kmi.open.ac.uk, accessed Dec-2019

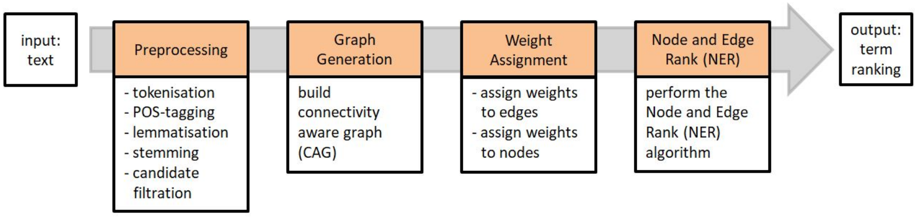


Fig. 1: An overview of Stage 1 (CoCoNoW) for automatic keyword detection

- We present a novel graph-based keyword detection approach that identifies representative words from a given text and assigns weights to rank them in the order of relevance.
- We also evaluated our proposed approach on three different publicly available datasets and consistently outperformed all other existing approaches.
- In this paper, we also complement our keyword detection system with ontology-based topic modeling to identify topics from a given publication.

The rest of the paper is structured as follows: Section 2 describes the methodology of the CoCoNoW approach and topic modeling. The performance of the keyword detection and topic modeling is evaluated in Section 3. Finally, the presented work is concluded in Section 4.

2 Methodology

This paper proposes a two-staged novel approach in which the first stage deals with automatic keyword detection called Collective Connectivity-Aware Node Weight (CoCoNoW) and in the second stage, the detected keywords are consolidated with the Computer Science Ontology [35] to identify topics for a given scientific publication. In CoCoNoW, we present a unique fusion of Collective Node Weight [6], Connectivity Aware Graphs (CAGs) [12] and Positional Weight [13] to identify keywords from a given document in order to cluster publications with common topics together. Details of the proposed approach are as follows:

2.1 Stage 1: Automatic keyword detection using CoCoNoW

Preprocessing CoCoNoW uses the standard preprocessing steps like tokenization, part of speech tagging, lemmatization, stemming and candidate filtration. A predefined list of stop words is used to identify stop words. There are several stop word lists available for the English language. For the sake of a fair evaluation and comparison, we selected the stopword list[4] used by the most recent approach by Duari and Bhatnagar [12]. Additionally, any words with less than three characters are considered stop words and are removed from the text.

[4] http://www.lextek.com/manuals/onix/stopwords2.html, accessed Dec 2019

CoCoNoW also introduces the Minimal Occurrence Frequency (MOF) which is inspired by average occurrence frequency (AOF) [6]. MOF can be represented as follows:

$$\mathrm{MOF}(D, \beta) = \beta \frac{\sum\limits_{t \in D} \mathrm{freq(t)}}{|D|} \tag{1}$$

where β is a parameter, $|D|$ is the number of terms in the document D and freq(t) is the frequency of term t. The MOF supports some variation with the parameter β; a higher β means more words get removed, whereas a lower β means fewer words get removed. This allows customizing the CoCoNoW to the document length: Longer Documents contain more words, therefore, having a higher frequency of terms. Parameter optimization techniques on various datasets suggest that the best values for β are about 0.5 for short documents e.g. only analyzing abstracts of papers rather than the entire text; and 0.8 for longer documents such as entire papers.

Graph Building CoCoNoW is a graph-based approach, it represents the text as a graph. We performed experiments with various window sizes for CoCoNoW, including different numbers of consecutive sentences for the dynamic window size employed by CAGs [12]. The performance dropped when more than two consecutive sentences were considered in one window. Therefore, a dynamic window size of two consecutive sentences was adopted for CoCoNoW. This means that an edge is added between any two terms if they occur within two consecutive sentences.

Weight Assignments CoCoNoW is based on the Keyword Extraction using Collective Node Weight (KECNW) model developed by Biswas et al. [6]. The general idea is to assign weights to the nodes and edges that incorporate many different features, such as frequency, centrality, position, and weight of the neighborhood.

Edge Weights The weight of an edge typically depends on the relationship it represents, in our case this relationship is term co-occurrence. Hence, the weight assigned to the edges is the normalized term co-occurrence $w(e)$, which is computed as follows:

$$w(e) = \frac{\mathrm{coocc}(t_u, t_v)}{\mathrm{maxCoocc}} \tag{2}$$

where the weight $w(e)$ of an edge $e = \{u, v\}$ is obtained by dividing the number of times the corresponding terms t_u and t_v co-occur in a sentence ($coocc(t_u, t_v)$) by the maximum number of times any two terms co-occur in a sentence ($maxCoocc$). This is essentially a normalization of the term co-occurrence.

Node Weights The final node weight is a summation of four different features. Two of these features, namely *distance to most central node* and *term frequency* are also used by [6]. In addition, we employed *positional weight* [13] and the newly introduced *summary bonus*. All of these features are explained as follows:

Distance to most central node: Let c be the node with the highest degree. This node is considered the most central node in the graph. Then assign the inverse distance $D_C(v)$ to this node as the weight for all nodes:

$$D_C(v) = \frac{1}{d(c,v)+1} \tag{3}$$

where $d(c,v)$ is the distance between node v and the most central node c.

Term Frequency: The number of times a term occurs in the document divided by the total number of terms in the document:

$$\mathrm{TF}(t) = \frac{\mathrm{freq}(t)}{|D|} \tag{4}$$

where freq(t) is the frequency of term t and $|D|$ is the total number of terms in the document D.

Summary Bonus: Words occurring in summaries of documents, e.g. abstracts of scientific articles, are likely to have a higher importance than words that only occur in rest of the document:

$$\mathrm{SB}(t) = \begin{cases} 0 & \text{if t does not occur in the summary} \\ 1 & \text{if t occurs in the summary} \end{cases} \tag{5}$$

where $\mathrm{SB}(t)$ is the summary bonus for term t. If there is no such summary, the summary bonus is set to 0.

Positional Weight: As proposed by Florescu and Caragea [13], words appearing in the beginning of the document have a higher chance of being important. The positional weight $PW(t)$ is based on this idea and is computed as follows:

$$\mathrm{PW}(t) = \sum_{j}^{\mathrm{freq}(t)} \frac{1}{p_j} \tag{6}$$

where freq(t) is the number of times term t occurs in the document and p_j is the position of the j^{th} occurrence in the text.

Final weight computation for CoCoNoW: The final node weight W uses all these features described above and combines them as follows:

$$W(v) = \mathrm{SB}(t_v) + D_C(v) + \mathrm{PW}(t_v) + \mathrm{TF}(t_v) \tag{7}$$

where t_v is the term corresponding to node v, SB is the summary bonus, D_C is the distance to the most central node, PW is the positional weight and TF is the term frequency. All individual summands have been normalized in the following way:

$$\mathrm{norm}(x) = \frac{x - \mathrm{minVal}}{\mathrm{maxVal} - \mathrm{minVal}} \tag{8}$$

where x is a feature for an individual node, minVal is the smallest value of this feature and maxVal is the highest value of this feature. With this normalization, each summand in equation 7 lies in the interval $[0, 1]$. Thus, all summands are considered to be equally important.

Node and Edge Rank (NER) Both the assigned node and edge weights are then used to perform Node and Edge Rank (NER) [5]. This is a variation of the famous PageRank [31] and is recursively computed as given below:

$$\mathrm{NER}(v) = (1-d)W(v) + dW(v) \sum_{e=(u,v)} \frac{w(e)}{\sum\limits_{e'=(u,w)} w(e')} \mathrm{NER}(u) \tag{9}$$

where d is the damping factor, which regulates the probability of jumping from one node to the next one [6]. The value for d is typically set to 0.85. $W(v)$ is the weight of node v as computed in equation 7. $w(e)$ denotes the edge weight of edge e, $\sum_{e'=(u,w)} w(e')$ denotes the summation over all weights of incident edges of an adjacent node u of v and NER(u) is the Node and Edge Rank of node u.

This recursion stops as soon as the absolute change in the NER value is less than the given threshold of 0.0001. Alternatively, the execution ends as soon as a total of 100 iterations are performed. However, it is just a precaution, as the approach usually converges in about 8 iterations. Mihalcea et al. [24] report that the approach needed about 20 to 30 steps to converge for their dataset. All nodes are then ranked according to their NER. Nodes with high values are more likely to be keywords. Each node corresponds to exactly one term in the document, so the result is a priority list of terms that are considered keywords.

2.2 Stage 2: Topic Modeling

In this section, we will discuss the second stage of our approach. The topic modeling task is increasingly popular on social web data [27, 3, 1, 37], where the topics of interest are unknown beforehand. However, this is not the case for the task in hand, i.e. clustering publications based on their topics. All publications share a common topic, for example, all ICDAR papers have *Document Analysis* as a common topic. Our proposed approach takes advantage of the common topic by incorporating an ontology. In this work, an ontology is used to define the possible topics where the detected keywords of each publication are subsequently mapped onto the topics defined by the ontology. For this task, we processed ICDAR publications from 1993 to 2017. The reason for selecting ICDAR publications for this task is that we already had the citation data available for these publications which will eventually be helpful during the evaluation of this task.

Topic Hierarchy Generation All ICDAR publications fall under the category of *Document Analysis*. The first step is to find a suitable ontology for the ICDAR publications. For this purpose, the CSO 3.1[5] [35] was employed. This ontology was built using the Klink-2 approach [29] on the Rexplore dataset [30] which contains about 16 million publications from different research areas in the field of computer science. These research areas are represented as the entities in the ontology. The reason for using this ontology rather than other manually crafted taxonomies is that it was extracted from publications with the latest topics that occur in publications. Furthermore, Salatino et al. [36] used this ontology already for the same task. They proposed an approach for the classification of research topics and used the CSO as a set of available classes. Their approach was based on bi-grams and tri-grams and computes the similarity of these to the nodes in the ontology by leveraging word embeddings from word2vec [25].

Computer Science Ontology The CSO 3.1 contains $23,800$ nodes and $162,121$ edges. The different relations between these nodes are based on the Simple Knowledge Organization System[6] and include eight different types of relations.

Hierarchy Generation For this task, we processed ICDAR publications from 1993 to 2017. Therefore, in line with the work of Breaux and Reed [8], the node *Document Analysis* is considered the root node for the ICDAR conference. This will be the root of the resulting hierarchy. Next, nodes are added to this hierarchy depending on their relations in the ontology. All nodes with the relation *superTopicOf* are added as children to the root. This continues recursively until there are no more nodes to add. Afterwards, three relation types *sameAs*, *relatedEquivalent* and *preferentialEquivalent* are used to merge nodes. The edges with these relations between terms describe the same concept, e.g. *optical character recognition* and *ocr*. One topic is selected as the main topic while all merged topics are added in the synonym attribute of that node. Note that all of these phrases are synonyms of essentially the same concept. The extracted keywords are later on matched against these sets of synonyms. Additionally, very abstract topics such as *information retrieval* were removed as they are very abstract and could potentially be a super-topic of most of the topics in the hierarchy thus making the hierarchy unnecessarily large and complicated. Lastly, to mitigate the missing topics of specialized topics like *Japanese Character Recognition*, we explored the official topics of interest for the ICDAR community[7]. An examination of

[5] https://cso.kmi.open.ac.uk, accessed Dec-2019

[6] https://www.w3.org/2004/02/skos, accessed Dec-2019

[7] https://icdar2019.org/call-for-papers, accessed Dec-2019

Table 1: Distribution details of datasets used

Dataset	|D|	L	Avg / SD	Dataset Description
Hulth2003 [14]	1500	129	19.5 / 9.98	Abstracts
NLM500 [2]	500	4854	23.8 / 8.19	Full papers
SemEval2010 [17]	244	8085	25.5 / 6.96	Full papers

the hierarchy revealed missing specialized topics like the only script dependent topic available in the hierarchy was *Chinese Characters*, so other scripts such as Greek, Japanese and Arabic were added as siblings of this node. We also created a default node labeled *miscellaneous* for all those specialized papers which can not be assigned to any of the available topics.

Eventually, the final topic hierarchy consists of 123 nodes and has 5 levels. The topics closer to the root are more abstract topics while the topics further away from the root node represent more specialized topics.

Topic Assignment Topics are assigned to papers by using two features of each paper: The title of the publication and the top 15 extracted keywords. The value of k=15 was chosen after manually inspecting the returned keywords; fewer keywords mean that some essential keywords are ignored, whereas a higher value means there are more unnecessary keywords that might lead to a wrong classification.

In order to assign a paper to a topic, we initialize the matching score with 0. The topics are represented as a set of synonyms, these are compared with the titles and keywords of the paper. If a synonym is a substring of the title, a constant of 200 is added to the score. The assumption is that if the title of a paper contains the name of a topic, then it is more likely to be a good candidate for that topic. Next, if all unigrams from a synonym are returned as keywords, the term frequency of all these unigrams is added to the matching score. By using a matching like this, different synonyms will have a different impact on the overall score depending on how often the individual words occurred in the text. To perform matching we used the Levenshtein distance [18] with a threshold of 1. This is the case to accommodate the potential plural terms. The constant bonus of 200 for a matching title comes from assessing the average document frequency of the terms. Most of the synonyms consist of two unigrams, so the document frequencies of two words are added to the score in case of a match. This is usually less than 200 - so the matching of the title is deemed more important.

Publications are assigned to 2.65 topics on average with a standard deviation of 1.74. However, the values of the assignments differ greatly between publications. The assignment score depends on the term frequency, which itself depends on the individual writing style of the authors. For this reason, the different matching scores are normalized: For each paper, we find the highest matching score, then we divide all matching scores by this highest value. This normalization means that every paper will have one topic that has a matching score of 1 - and the scores of other assigned topics will lie in the interval $(0, 1]$. This accounts for different term frequencies and thus, also the different writing styles.

3 Evaluation

In this section, we will discuss the experimental setup and the evaluation of our system where we firstly discuss the evaluation of our first stage CoCoNoW for keyword detection. The results from CoCoNoW are compared with various state-of-the-art approaches on three different datasets: Hulth2003 [14], SemEval2010 [17] and NLM500 [2]. Afterwards, we will discuss the evaluation results of the second stage for topic modeling as well.

3.1 Experimental Setup

Keyword detection approaches usually return a ranking of individual keywords. Hence, the evaluation is based on individual keywords. For the evaluation of these rankings, a parameter k is introduced where only the top k keywords of the rankings are considered. This is a standard procedure to evaluate performance [12, 17, 20, 22, 38].

However, as the gold-standard keywords lists contain key phrases, these lists undergo a few preprocessing steps. Firstly, the words are lemmatized and stemmed, then a set of strings called the evaluation set is created. It contains all unigrams. All keywords occur only once in the set, and the preprocessing steps allow the matching of similar words with different inflections. The top k returned keywords are compared with this evaluation set.

Note that the evaluation set can still contain words that do not occur in the original document, which is why an F-Measure of 100% is infeasible. For example, the highest possible F-Measure for the SemEval2010 dataset is only 81% because 19% of the gold standard keywords do not appear in the corresponding text [17].

Table 1 gives an overview of performance on different datasets. These datasets were chosen because they cover different document lengths, ranging from about 130 to over 8000 words and belong to different domains: biomedicine, information technology, and engineering.

3.2 Performance Evaluation Stage 1: CoCoNoW

The performance of keyword detection approaches is assessed by matching the top k returned keywords with the set of gold standard keywords. The choice of k influences the performance of all keyword detection approaches as the returned ranking of keywords differs between these approaches. Table 2 shows the performance in terms of Precision, Recall and F-Measure of several approaches for different values of k. Fig. 2 and Fig. 3 compare the performance of our approach

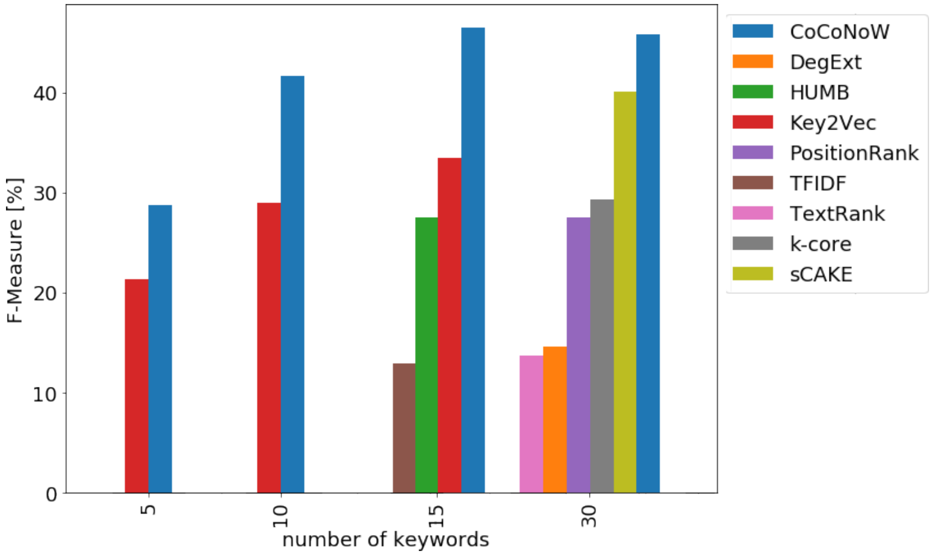


Fig. 2: Evaluation of CoCoNoW on the SemEval2010 [17] dataset.

with other approaches on the **SemEval2010** [17] dataset and the **Hulth2003** [14] dataset respectively.

By looking at the results of all trials, we can make the following observations:

- CoCoNoW always has the highest F-measure
- CoCoNoW always has the highest Precision
- In the majority of the cases, CoCoNoW also has the highest Recall

For the **Hulth2003** [14] dataset, CoCoNoW achieved the highest F-measure of 57.2% which is about 6.8% more than the previous state-of-the-art. On the **SemEval2010** [17] dataset, CoCoNoW achieved the highest F-measure of 46.8% which is about 6.2% more than the previous state-of-the-art. Lastly, on the **NLM500** [2] dataset, CoCoNoW achieved the highest F-measure of 29.5% which is about 1.2% better than previous best performing approach. For $k = 5$ CoCoNoW achieved the same Recall as the Key2Vec approach by Mahata et al. [22], however, the Precision was 15.2% higher. For $k = 10$, the approach by Wang et al. [38] had a Recall of 52.8%, whereas CoCoNoW only achieved 41.9%; this results in a difference of 10.9%. However, the Precision of CoCoNoW is almost twice as high, i.e. 73.3% as compared to 38.7%. Lastly, for $k = 25$, the sCAKE algorithm by Duari and Bhatnagar [12] has a higher Recall (0.6% more), but also a lower Precision (9.4% less). All in all, CoCoNoW extracted the most keywords successfully and outperformed other state-of-the-art approaches. Note the consistently high Precision values of CoCoNoW: There is a low number of false positives (i.e. words wrongly marked as keywords), which is crucial in the next stage of clustering publications with respect to their respective topics.

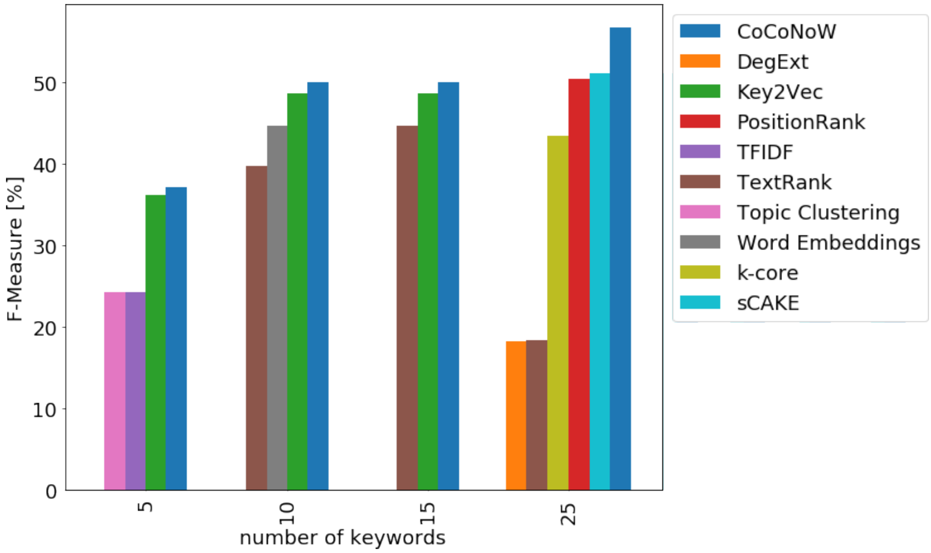


Fig. 3: Evaluation of CoCoNoW on the Hulth2003 [14] dataset.

3.3 Performance Evaluation Stage 2: Topic Modeling

This section discusses the evaluations of ontology-based topic modeling. There was no ground truth available for the ICDAR publications, consequently making the evaluation of topic modeling a challenging task. Nevertheless, we employed two different approaches for evaluation: manual inspection and citation count. Details of both evaluations are as follows:

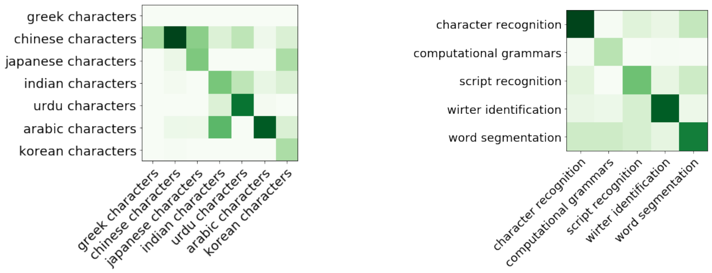
Manual Inspection The proposed method for topic assignment comes with labels for the topics, so manually inspecting the papers assigned to a topic is rather convenient. This is done by going through the titles of all papers assigned to a topic and judging whether the assignment makes sense.

For specialized topics i.e. the ones far from the root, the method worked very well, as it is easy to identify papers that do not belong to a topic. Manual inspection showed that there are very few false positives, i.e. publications assigned to an irrelevant topic. This is because of the high Precision of the CoCoNoW algorithm: The low number of false positives in the extracted keywords increases the quality of the topic assignment. The closer a topic is to the root i.e. a more generic topic, the more difficult it is to assess whether a paper should be assigned to it: Often, it is not possible to decide whether a paper can be assigned to a general node such as *neural networks* by just reading the title. Hence, this method does not give meaningful results for more general topics. Furthermore, this method was only able to identify false positives. It is difficult to identify false negatives with this method, i.e. publications not assigned to relevant topics.

Table 2: Performance comparison of CoCoNoW with several other approaches on the different datasets.
* Results reported by Duari and Bhatnagar [12], not by the original authors.

Approach	**k**	**Hulth2003** [14]			**SemEval2010** [17]			**NLM500** [2]		
		P[%]	**R**[%]	**F1**[%]	**P**[%]	**R**[%]	**F1**[%]	**P**[%]	**R**[%]	**F1**[%]
TF-IDF [20]	5	33.3	17.3	24.2	-	-	-	-	-	-
Topic Clustering [20]		35.4	18.3	24.3	-	-	-	-	-	-
Key2Vec [22]		68.8	**25.7**	36.2	41.0	14.4	21.3	-	-	-
CoCoNoW		**84.0**	**25.7**	**37.3**	**84.1**	**17.5**	**28.7**	48.8	11.4	17.9
TextRank [24]	10	45.4	47.1	39.8	-	-	-	-	-	-
Word Embeddings [38]		38.7	**52.8**	44.7	-	-	-	-	-	-
Key2Vec [22]		57.6	42.0	48.6	35.3	24.7	29.0	-	-	-
CoCoNoW		**73.3**	41.9	**50.0**	**72.3**	**29.8**	**41.6**	43.3	19.8	26.3
supervised approach [14]	16	25.2	51.7	33.9	-	-	-	-	-	-
TextRank [24]	14	31.2	43.1	36.2	-	-	-	-	-	-
TF-IDF [17]	15	-	-	-	11.6	14.5	12.9	-	-	-
HUMB [21]	15	-	-	-	27.2	27.8	27.5	-	-	-
Key2Vec [22]	15	55.9	50.0	52.9	34.4	32.5	33.4	-	-	-
CoCoNoW	15	**63.5**	**52.9**	**54.2**	**62.2**	**39.2**	**46.5**	37.11	25.2	29.0
TextRank [24]	25	-	-	18.4*	-	-	-	-	-	-
DegExt [19]		-	-	18.2*	-	-	-	-	-	-
k-core [34]		-	-	43.4*	-	-	-	-	-	-
PositionRank [13]		45.7*	64.5*	50.4*	-	-	-	-	-	-
sCAKE [12]		45.4	**66.8**	51.1	-	-	-	-	-	-
CoCoNoW		**54.8**	66.2	**56.8**	47.3	47.8	46.8	29.3	32.6	29.9
TextRank [24]	30	-	-	-	-	-	13.7*	-	-	10.7*
DegExt [19]		-	-	-	-	-	14.6*	-	-	10.9*
k-core [34]		-	-	-	-	-	29.3*	-	-	20.2*
PositionRank [13]		-	-	-	25.3*	31.3*	27.5*	19.7*	26.6*	21.9*
sCAKE [12]		-	-	-	35.8	47.4	40.1	24.5	35.0	28.3
CoCoNoW		52.5	70.1	57.2	**42.6**	**51.5**	**45.8**	**26.7**	**35.3**	**29.5**

Evaluation by Citation Count The manual inspection indicated that the topic assignment works well, but that was just a qualitative evaluation. So a second evaluation is performed. It is based on the following assumption: *Papers dealing with a topic cite other papers from the same topic more often than papers dealing with different topics.* We believe that this is a sensible assumption, as papers often compare their results with previous approaches that tackled similar problems. So instead of evaluating the topic assignment directly, it is indirectly evaluated be counting the number of citations between the papers assigned to the topics. However, a paper can be assigned to multiple overlapping topics (e.g. *machine learning* and *neural networks*). This makes it infeasible to compare topics at two different hierarchy levels using this method.



(a) *Character Recognition* (b) *Pattern Recognition*

Fig. 4: Citation count for different super topics.

Nevertheless, this method is suited for evaluating the topic assignment of siblings, i.e. topics that have the same super topic. Figure 4 shows heatmaps for citations between child elements of a common supertopic in different levels in the hierarchy. The rows represent the number of citing papers, the columns the number of cited papers. A darker color in a cell represents more citations. Fig. 4a shows the script-dependent topics, which are rather specialized and all have the common supertopic *character recognition*. The dark diagonal values clearly indicate that the number of intra-topic citations are higher than inter-topic citations.

Figure 4b shows the subtopics of the node *pattern recognition*, which has a very high level of abstraction and is close to the root of the hierarchy. The cells in the diagonal are clearly the darkest ones again, i.e. there are more citations within the same topic than between different topics. This is a recurring pattern across the entire hierarchy. So in general, results from these evaluations indicate that the topic assignment works reliably if our assumption is correct.

4 Conclusion

This paper presents a two-staged novel approach where the first stage called CoCoNoW deals with automatic keyword detection while the second stage uses the detected keywords and performs ontology-based topic modeling on a given scientific publication. Evaluations of CoCoNoW clearly depict its supremacy over several existing keyword detection approaches by consistently outperforming every single approach in terms of Precision and F-Measure. On the other hand, the evaluation of the topic modeling approach suggests that it is an effective technique as it uses an ontology to accurately define context and domain. Hence, the publications with the same topics can be clustered more reliably.

References

1. Aiello, L.M., Petkos, G., Martin, C., Corney, D., Papadopoulos, S., Skraba, R., Göker, A., Kompatsiaris, I., Jaimes, A.: Sensing trending topics in twitter. IEEE Transactions on Multimedia **15**(6), 1268–1282 (2013)

2. Aronson, A.R., Bodenreider, O., Chang, H.F., Humphrey, S.M., Mork, J.G., Nelson, S.J., Rindflesch, T.C., Wilbur, W.J.: The nlm indexing initiative. In: Proceedings of the AMIA Symposium. p. 17. American Medical Informatics Association (2000)
3. Becker, H., Naaman, M., Gravano, L.: Beyond trending topics: Real world event identification on twitter. In: AAAI (2011)
4. Beliga, S.: Keyword extraction: a review of methods and approaches. University of Rijeka, Department of Informatics pp. 1–9 (2014)
5. Bellaachia, A., Al-Dhelaan, M.: Ne-rank: A novel graph-based keyphrase extraction in twitter. In: 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology. vol. 1, pp. 372–379. IEEE (2012)
6. Biswas, S.K., Bordoloi, M., Shreya, J.: A graph based keyword extraction model using collective node weight. Expert Systems with Applications **97**, 51–59 (2018)
7. Boudin, F.: Unsupervised keyphrase extraction with multipartite graphs. arXiv preprint arXiv:1803.08721 (2018)
8. Breaux, T.D., Reed, J.W.: Using ontology in hierarchical information clustering. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences. pp. 111b–111b. IEEE (2005)
9. Carpena, P., Bernaola-Galván, P., Hackenberg, M., Coronado, A., Oliver, J.: Level statistics of words: Finding keywords in literary texts and symbolic sequences. Physical Review E **79**(3), 035102 (2009)
10. Carretero-Campos, C., Bernaola-Galván, P., Coronado, A., Carpena, P.: Improving statistical keyword detection in short texts: Entropic and clustering approaches. Physica A: Statistical Mechanics and its Applications **392**(6), 1481–1492 (2013)
11. Carston, R.: Linguistic communication and the semantics/pragmatics distinction. Synthese **165**(3), 321–345 (2008)
12. Duari, S., Bhatnagar, V.: scake: Semantic connectivity aware keyword extraction. Information Sciences **477**, 100–117 (2019)
13. Florescu, C., Caragea, C.: A position-biased pagerank algorithm for keyphrase extraction. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
14. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 conference on Empirical methods in natural language processing. pp. 216–223. Association for Computational Linguistics (2003)
15. Johnson, R., Watkinson, A., Mabe, M.: The stm report. Tech. rep., International Association of Scientific, Technical, and Medical Publishers (2018)
16. Kecskés, I., Horn, L.R.: Explorations in pragmatics: Linguistic, cognitive and intercultural aspects, vol. 1. Walter de Gruyter (2008)
17. Kim, S.N., Medelyan, O., Kan, M.Y., Baldwin, T.: Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In: Proceedings of the 5th International Workshop on Semantic Evaluation. pp. 21–26 (2010)
18. Levenshtein, V.I.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics Doklady **10**, 707 (Feb 1966)
19. Litvak, M., Last, M., Aizenman, H., Gobits, I., Kandel, A.: Degext—a language-independent graph-based keyphrase extractor. In: Advances in Intelligent Web Mastering–3, pp. 121–130. Springer (2011)
20. Liu, Z., Li, P., Zheng, Y., Sun, M.: Clustering to find exemplar terms for keyphrase extraction. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1. pp. 257–266 (2009)
21. Lopez, P., Romary, L.: Humb: Automatic key term extraction from scientific articles in grobid. In: Proceedings of the 5th international workshop on semantic evaluation. pp. 248–251. Association for Computational Linguistics (2010)

22. Mahata, D., Shah, R.R., Kuriakose, J., Zimmermann, R., Talburt, J.R.: Theme-weighted ranking of keywords from text documents using phrase embeddings. In: 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR). pp. 184–189. IEEE (2018). https://doi.org/10.31219/osf.io/tkvap
23. Mey, J.L.: Whose language?: a study in linguistic pragmatics, vol. 3. John Benjamins Publishing (1985)
24. Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing (2004)
25. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space. CoRR **abs/1301.3781** (2013)
26. Nikolentzos, G., Meladianos, P., Stavrakas, Y., Vazirgiannis, M.: K-clique-graphs for dense subgraph discovery. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 617–633. Springer (2017)
27. O'Connor, B., Krieger, M., Ahn, D.: Tweetmotif: Exploratory search and topic summarization for twitter. In: AAAI (2010)
28. Ohsawa, Y., Benson, N.E., Yachida, M.: Keygraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In: Proceedings IEEE International Forum on Research and Technology Advances in Digital Libraries-ADL'98-. pp. 12–18. IEEE (1998). https://doi.org/10.1109/adl.1998.670375
29. Osborne, F., Motta, E.: Klink-2: integrating multiple web sources to generate semantic topic networks. In: ISWC. pp. 408–424. Springer (2015)
30. Osborne, F., Motta, E., Mulholland, P.: Exploring scholarly data with rexplore. In: International semantic web conference. pp. 460–477. Springer (2013)
31. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Tech. rep., Stanford InfoLab (1999)
32. Pay, T., Lucci, S.: Automatic keyword extraction: An ensemble method. In: Conference: IEEE Big Data 2017, At Boston (12 2017)
33. Rabby, G., Azad, S., Mahmud, M., Zamli, K.Z., Rahman, M.M.: A flexible keyphrase extraction technique for academic literature. Procedia Computer Science **135**, 553–563 (2018)
34. Rousseau, F., Vazirgiannis, M.: Main core retention on graph-of-words for single-document keyword extraction. In: European Conference on Information Retrieval. pp. 382–393. Springer (2015)
35. Salatino, A., Thanapalasingam, T., Mannocci, A., Osborne, F., Motta, E.: The Computer Science Ontology: A Large-Scale Taxonomy of Research Areas: 17th International Semantic Web Conference, Monterey, CA, USA, October 8–12, 2018, Proceedings, Part II, pp. 187–205. Springer (01 2018)
36. Salatino, A.A., Osborne, F., Thanapalasingam, T., Motta, E.: The cso classifier: Ontology-driven detection of research topics in scholarly articles. In: Doucet, A., Isaac, A., Golub, K., Aalberg, T., Jatowt, A. (eds.) Digital Libraries for Open Knowledge. pp. 296–311. Springer International Publishing, Cham (2019)
37. Slabbekoorn, K., Noro, T., Tokuda, T.: Ontology-assisted discovery of hierarchical topic clusters on the social web. J. Web Eng. **15**(5&6), 361–396 (2016)
38. Wang, R., Liu, W., McDonald, C.: Using word embeddings to enhance keyword identification for scientific publications. In: ADC. pp. 257–268. Springer (2015)
39. Ware, M., Mabe, M.: The stm report: An overview of scientific and scholarly journal publishing. Tech. rep., International Association of Scientific, Technical, and Medical Publishers (2015)