



HAL
open science

Large-Scale Evaluation of Keyphrase Extraction Models

Ygor Gallina, Florian Boudin, Béatrice Daille

► **To cite this version:**

Ygor Gallina, Florian Boudin, Béatrice Daille. Large-Scale Evaluation of Keyphrase Extraction Models. ACM/IEEE Joint Conference on Digital Libraries (JCDL), Aug 2020, Wuhan, China. 10.1145/1122445.1122456 . hal-02878953

HAL Id: hal-02878953

<https://hal.science/hal-02878953v1>

Submitted on 23 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Large-Scale Evaluation of Keyphrase Extraction Models

Ygor Gallina
LS2N, Université de Nantes
Nantes, France
ygor.gallina@univ-nantes.fr

Florian Boudin
LS2N, Université de Nantes
Nantes, France
florian.boudin@univ-nantes.fr

Béatrice Daille
LS2N, Université de Nantes
Nantes, France
beatrice.daille@univ-nantes.fr

ABSTRACT

Keyphrase extraction models are usually evaluated under different, not directly comparable, experimental setups. As a result, it remains unclear how well proposed models actually perform, and how they compare to each other. In this work, we address this issue by presenting a systematic large-scale analysis of state-of-the-art keyphrase extraction models involving multiple benchmark datasets from various sources and domains. Our main results reveal that state-of-the-art models are in fact still challenged by simple baselines on some datasets. We also present new insights about the impact of using author- or reader-assigned keyphrases as a proxy for gold standard, and give recommendations for strong baselines and reliable benchmark datasets.

CCS CONCEPTS

• **Information systems** → **Digital libraries and archives; Information retrieval**; • **Computing methodologies** → **Information extraction**.

KEYWORDS

Keyphrase generation; natural language processing; evaluation

ACM Reference Format:

Ygor Gallina, Florian Boudin, and Béatrice Daille. 2020. Large-Scale Evaluation of Keyphrase Extraction Models. In *ACM/IEEE Joint Conference on Digital Libraries in 2020 (JCDL '20), August 1–5, 2020, Virtual Event, China*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

Keyphrases are single or multi-word lexical units that represent the main concepts in a document [13]. They are particularly useful for indexing, searching and browsing digital libraries [3, 20, 46, 50], and have proven themselves as effective features in many downstream natural language processing tasks [5, 23, 28]. Still, most documents do not have assigned keyphrases, and manual annotation is simply not a feasible option [31]. There is therefore a great need for automated methods to assign relevant keyphrases to documents.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

JCDL '20, August 1–5, 2020, Virtual Event, China

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7585-6/20/08...\$15.00

<https://doi.org/10.1145/1122445.1122456>

Automatic keyphrase extraction¹ – that is, the task of extracting keyphrases either from the content of the document or from a controlled vocabulary – has received much attention from the research community [1, 18, 26]. Thus, many keyphrase extraction models were proposed over the last years, ranging from early statistics-based models [47], to popular graph-based ranking models [35], and recent neural models [34]. However, because of the great discrepancies in experimental setups among past studies, it is very difficult to compare and contrast the effectiveness of these models, and even more so to assess the progress of the field as a whole.

More specifically, we observe striking differences in how models are parameterized, evaluated and compared in previous work. To name just a few examples, experiments are most often conducted on different benchmark datasets, all of which differ in domain, size, language or quality of the gold standard (that is, reference keyphrases supplied by authors, readers or professional indexers). This not only makes the reported results hard to contrast, but also has a profound impact on trained model performance [15]. In addition, and since there is no consensus as to which evaluation metric is most reliable for keyphrase extraction [21, 24, 49], diverse measures are commonly seen in the literature, thus preventing any further direct comparisons. Moreover, the evaluation of missing keyphrases – that is, gold keyphrases that do not occur in the content of the document – is still an open question and there is little agreement on whether they should be included or not [26].

We strongly believe that this lack of empirical rigor is a real hindrance to progress on keyphrase extraction, and that a systematic comparison of existing models under the same conditions is needed to fully understand how they actually perform. In this work, we resolve this issue by conducting the first large-scale study on automatic keyphrase extraction. More precisely, we present an extensive comparative analysis of state-of-the-art keyphrase extraction models involving 9 benchmark datasets from various domains. To ensure controlled, fair and reliable experiments, we embarked upon the difficult process of re-implementing all of the models presented in this paper² and pre-processing the datasets in a unified and systematic way³.

Using these new large-scale experimental results, we seek to better understand how well state-of-the-art models perform across sources, domains and languages. We also go further than prior work and investigate the following research questions:

- (1) How much progress have we made on keyphrase extraction since early models?
- (2) What is the impact of using non-expert gold standards, that is, author- or reader-assigned keyphrases, when training and evaluating keyphrase extraction models?

¹Also referred to as keyphrase generation or keyphrase annotation.

²<https://github.com/boudinfl/pke>

³<https://github.com/boudinfl/ake-datasets>

- (3) Which baselines and benchmark datasets should be included in future work for a better understanding of the pros and cons of a newly proposed model?

2 BENCHMARK DATASETS

Benchmark datasets for evaluating automatic keyphrase extraction cover a wide range of sources ranging from scientific articles and web pages to twitter and email messages. We collected 9 of the most widely used datasets which we believe are representative of the different sources and domains found in previous work. Detailed statistics for each selected dataset are shown in Table 1. They are grouped into three categories that are outlined below:

Scientific articles Among the selected datasets, three are composed of full-text scientific publications: ACM [27] and SemEval [26] about computer science, and PubMed [38] from the medical domain. Not surprisingly, they contain only a small number of documents due to copyright reasons. These datasets provide author-assigned keyphrases which serve as a reasonable, but far from perfect, proxy for expert annotations. In the case of SemEval, student annotators were hired to extend gold annotation labels.

Paper abstracts Scientific abstracts, often referred to as bibliographic records, are arguably the most prevalent documents for benchmarking keyphrase extraction. They are readily available in great quantities and come with author-assigned keyphrases that can be used as gold standard. We gathered three datasets, all dealing with the computer science domain: Inspec [22], WWW [9] and KP20k [34]. It is worth noting that with more than half a million documents, KP20k is the largest dataset to date and one of the few that is large enough to train neural models.

News articles News texts are the last source of documents present among the collected datasets. Similar to paper abstracts, online news are available in large quantities and can be easily mined from the internet. We selected the following three datasets: DUC-2001 [43], 500N-KPCrowd [33] and KP-Times [15]. The first two datasets provide reader-assigned keyphrases, while KPTimes supplies indexer-assigned keyphrases extracted from metadata and initially intended for search engines. It is interesting to observe that only two datasets in our study, namely Inspec and KPTimes, provide gold keyphrases annotated by professional indexers.

Datasets containing scientific articles or abstracts rely primarily on author-assigned keyphrases as gold standard. They therefore exhibit similar properties for the average number of ground truth keyphrases per document (≈ 5). On the other hand, articles are on average significantly longer than abstracts (≈ 7500 words vs. ≈ 160 words respectively) and consequently reveal a much smaller fraction of missing keyphrases ($\approx 18\%$ vs. $\approx 39\%$ respectively). Datasets with reader-assigned keyphrases exhibit the lowest numbers of missing keyphrases, which can be explained by the fact that readers appear to produce gold-standard annotations in an extractive fashion [45]. We also confirmed this empirically by computing the ratio of missing keyphrases in the author-assigned (24%) and reader-assigned (17.5%) gold annotations of the SemEval dataset.

Dataset	Ann.	Train	Test	#words	#kp	%abs
PubMed [38]	A	-	1 320	5 323	5.4	16.9
ACM [27]	A	-	2 304	9 198	5.3	16.3
SemEval [26]	A \cup R	144	100	7 961	14.7	19.7
<i>Scientific articles (avg.)</i>				7 494	8.5	17.6
Inspec [22]	I	1 000	500	135	9.8	22.4
WWW [9]	A	-	1 330	164	4.8	52.0
KP20k [34]	A	530K	20K	176	5.3	42.6
<i>Paper abstracts (avg.)</i>				158	6.6	39.0
DUC-2001 [43]	R	-	308	847	8.1	3.7
KPCrowd [33]	R	450	50	465	46.2	11.2
KPTimes [15]	I	260K	10K	921	5.0	54.7
<i>News articles (avg.)</i>				744	19.8	23.2

Table 1: Statistics of the datasets. Gold annotation is supplied by authors (A), readers (R) or professional indexers (I). The number of documents in the training and testing splits are shown. The average number of keyphrases (#kp) and words (#words) per document, and the ratio of missing keyphrases (%abs) are computed on the test set.

In contrast, the opposite trend is observed for KPTimes that comes with gold standards annotated by professional indexers and that shows the highest percentage of missing keyphrases (54.7%). This indicates the more abstractive nature of indexer-assigned keyphrases. Put differently, it is known that non-expert annotations are less constrained and may include seldom-used variants or misspellings [39], whereas indexers strive to rely on a consistent terminology and assign the same keyphrase to all documents for a given topic, even when it does not occur in these documents.

To investigate this further, we looked at how many variants of an index term, in this case “*artificial neural network*”, could be found in the author-assigned keyphrases of KP20k. All in all, we found dozens of variants for this term, including “*neural network*”, “*neural network (nns)*”, “*neural net*”, “*artificial neural net*” or “*nn*”. This apparent lack of annotation consistency intuitively has two consequences: 1) it makes it harder for supervised approaches to learn a good model, 2) it makes automatic evaluation much less reliable as it is based on exact string matching.

It is important to stress that datasets containing scientific articles may contain noisy texts. Indeed, most articles were automatically converted from PDF format to plain text and thus are likely to contain irrelevant pieces of text (e.g. muddled sentences, equations). Previous work show that noisy inputs undermine the overall performance of keyphrase extraction models [8]. In this study, we do not insist on a perfect input and we are aware that reported results may be improved with an increase in pre-processing effort.

3 MODELS

Roughly speaking, previous works on keyphrase extraction can be divided into two groups depending on whether they adopt a supervised learning procedure or not. This section starts by introducing

the baselines we will use in our experiments, and then proceeds to describe the state-of-the-art keyphrase extraction models we re-implemented sorted into the aforementioned two groups.

3.1 Baselines

Having strong baselines to compare with is a prerequisite for contrasting the results of proposed models. In previous studies, various baselines were considered, complicating the analysis and interpretation of the reported results. Our stance here is to establish three baselines, each associated with a particular feature that is commonly used in keyphrase extraction models. All baselines are also unsupervised, allowing their use and performance analysis on any of the benchmark datasets

Keyphrase position is a strong signal for both unsupervised and supervised models, simply because texts are usually written so that the most important ideas go first [32]. In single document summarization for example, the lead baseline –that is, the first sentences from the document–, while incredibly simple, is still a competitive baseline [25]. Similar to the lead baseline, we propose the **First-Phrases** baseline that extracts the first N keyphrase candidates from a document. We are not aware of any previous work reporting that baseline, yet, as we will see in §5, it achieves remarkably good results.

Graph-based ranking models for keyphrase extraction are, perhaps, the most popular models in the literature. Therefore, as a second baseline, we use **TextRank** [35], which weights keyphrase candidates using a random walk over a word-graph representation of the document. In a nutshell, TextRank defines the importance of a word in terms of how it relates to other words in the document, and ranks candidates according to the words they contain.

The third baseline, **TF×IDF** [37], have been repeatedly used in previous comparative studies [26, 34, *inter alia*]. In contrast with the other two baselines that do not require any resources whatsoever (beyond the document itself), TF×IDF makes use of the statistics collected from unlabelled data to weight keyphrase candidates. As such, it often gives better results, in some cases even on par with state-of-the-art models [48].

3.2 Unsupervised models

Annotated data are not always available or easy to obtain, which motivates the further development of unsupervised models for keyphrase extraction. Besides, looking back at previous work, most attempts to address this problem employ unsupervised approaches. In this study, we selected three recent state-of-the-art models based on their reported performance.

The first model we investigate is **PositionRank** [14], a graph-based model that incorporates two features (position and frequency) into a biased PageRank algorithm. This model operates at the word level, and assigns a score to each candidate using the sum of its individual word scores. As such, it suffers from over-generation errors⁴ [21], but still achieves good performance on short texts.

The second model we consider, **MPRank** [7], relies on a multipartite graph representation to enforce topical diversity while

⁴These errors occur when a model correctly outputs a keyphrase because it contains an important word, but at the same time erroneously predicts other keyphrases because they contain the same word.

ranking keyphrase candidates. It includes a mechanism to incorporate keyphrase selection preferences in order to introduce a bias towards candidates occurring first in the document. MultipartiteRank was shown to consistently outperform other unsupervised graph-based ranking models.

Both aforementioned models only exploit the document itself to extract keyphrases. The third model we include, **EmbedRank** [4], leverages sentence embeddings for ranking keyphrase candidates. Candidates are weighted according to their cosine distance to the document embedding, while diversity in the selected keyphrases is promoted using Maximal Marginal Relevance (MMR) [17]. Despite its simplicity, this model was shown to outperform other unsupervised models on short texts (abstracts and news).

3.3 Supervised models

Supervised models can be further divided into two categories, depending on whether they rely on a neural network or not.

Traditional supervised models treat the keyphrase extraction problem as a binary classification task. Here, we include such a model, namely **Kea** [47], in order to precisely quantify the performance gap with recent neural-based models. KEA uses a Naive Bayes classifier trained on a set of only two handcrafted features we have elected as baseline features: the TF×IDF score of the candidate and the normalized position of its first occurrence in the document. Previous work has reported confusing and conflicting results⁵ for Kea, raising questions about how it actually performs.

Neural models for keyphrase extraction rely on an encoder-decoder architecture [11, 40] with an attention mechanism [2, 29]. Training these models require large amounts of annotated training data, and is therefore only possible on the KP20k and KPTime datasets. The second supervised model we include in this study is **CopyRNN** [34], an encoder-decoder model that incorporates a copying mechanism [19] in order to be able to predict phrases that rarely occur. When properly trained, this model was shown to be very effective in extracting keyphrases from scientific abstracts.

The third supervised model we use, **CorrRNN** [10], extends the aforementioned model by introducing correlation constraints. It employs a coverage mechanism [42] that diversifies attention distributions to increase topic coverage, and a review mechanism to avoid generating duplicates. As such, it produces more diverse and less redundant keyphrases.

Note that only neural models have the ability to generate missing keyphrases, which in theory gives them a clear advantage over the other models.

4 EXPERIMENTAL SETTINGS

In addition to the variation in the choice of benchmark datasets and baselines, there are also major discrepancies in parameter settings and evaluation metrics between previous studies. For example, there is no point in contrasting the results in [34], [14] and [41], three papers about keyphrase extraction published in the same year at ACL, since neither benchmark datasets, parameter settings nor evaluation metrics are comparable. To address this problem, we use the same pre-processing tools, parameter settings and evaluation procedure across all our experiments.

⁵On SemEval, [34] report an F@10 score of 2.6 while [6] report a score of 19.3.

Model	Scientific articles						Paper abstracts						News articles					
	PubMed		ACM		SemEval		Inspec		WWW		KP20k		DUC-2001		KPCrowd		KPTimes	
	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP	F@10	MAP
FirstPhrases	15.4	14.7	13.6	13.5	13.8	10.5	29.3	27.9	10.2	9.8	13.5	12.6	24.6	22.3	17.1	16.5	9.2	8.4
TextRank	1.8	1.8	2.5	2.4	3.5	2.3	35.8	31.4	8.4	5.6	10.2	7.4	21.5	19.4	7.1	9.5	2.7	2.5
TF×IDF	16.7	16.9	12.1	11.4	17.7	12.7	36.5	34.4	9.3	10.1	11.6	12.3	23.3	21.6	16.9	15.8	9.6	9.4
PositionRank	4.9	4.6	5.7	4.9	6.8	4.1	34.2	32.2	11.6 [†]	8.4	14.1 [†]	11.2	28.6 [†]	28.0[†]	13.4	12.7	8.5	6.6
MPRank	15.8	15.0	11.6	11.0	14.3	10.6	30.5	29.0	10.8 [†]	10.4	13.6 [†]	13.3 [†]	25.6	24.9 [†]	18.2	17.0	11.2 [†]	10.1 [†]
EmbedRank	3.7	3.2	2.1	2.1	2.5	2.0	35.6	32.5	10.7 [†]	7.7	12.4	10.0	29.5[†]	27.5 [†]	12.4	12.4	4.0	3.3
Kea	18.6 [†]	18.6 [†]	14.2 [†]	13.3	19.5 [†]	14.7[†]	34.5	33.2	11.0 [†]	10.9 [†]	14.0 [†]	13.8 [†]	26.5 [†]	24.5 [†]	17.3	16.7	11.0 [†]	10.8 [†]
CopyRNN	24.2[†]	25.4[†]	24.4[†]	26.3[†]	20.3[†]	13.8	28.2	26.4	22.2[†]	24.9[†]	25.4[†]	28.7[†]	10.5	7.2	8.4	4.2	39.3[†]	50.9[†]
CorrRNN	20.8 [†]	19.4 [†]	21.1 [†]	20.5 [†]	19.4	10.9	27.9	23.6	19.9 [†]	20.3 [†]	21.8 [†]	22.7	10.5	6.5	7.8	3.2	20.5 [†]	20.3 [†]

Table 2: Performance of keyphrase extraction models. [†] indicates significance over the baselines.

4.1 Parameter settings

We pre-process all the texts using the Stanford CoreNLP suite [30] for tokenization, sentence splitting and part-of-speech (POS) tagging. All non-neural models operate on a set of keyphrase candidates, extracted from the input document. Selecting appropriate candidates is particularly important since it determines the upper bound on recall, and the amount of irrelevant candidates that models will have to deal with. For a fair and meaningful comparison, we use the same candidate selection heuristic across models. We follow the recommendation by Wang et al. [44] and select the sequences of adjacent nouns with one or more preceding adjectives of length up to five words. Candidates are further filtered by removing those shorter than 3 characters or containing non-alphanumeric symbols.

We implemented the neural models in PyTorch [36] using AllenNLP [16], and the non-neural models using the pke toolkit [6]. As neural models require large amounts of annotated data to be trained, we trained our models on the KP20k dataset for both scientific papers and abstracts, and on KPTimes for news texts. We compute Document Frequency (DF) counts and learn Kea models on training sets. For datasets without training splits, we apply a leave-one-out cross-validation procedure on the test sets for calculating DF counts and training models. We use the optimal parameters suggested by the authors for each model, and leverage pre-trained sentence embeddings⁶ for EmbedRank. We also found out that the training set of KP20k contains a non-negligible number of documents from the test sets of other datasets. We removed those documents prior to training.

4.2 Evaluation metrics

Although there is no consensus as to which metric is the most reliable for keyphrase extraction, a popular evaluation strategy is to compare the top k extracted keyphrases against the gold standard. We adopt this strategy and report the f-measure at the top 10 extracted keyphrases. In previous work, we often see differences in how gold standards are handled during evaluation. For example, some studies evaluate their models on the present and missing portions of the gold standard separately [10, 34, 48, *inter alia*], whereas other work use the entire gold standard [7, 14, *inter*

alia]. We chose the latter because recent models, in addition to extracting keyphrases from the content of the document, are able to generate missing keyphrases. Following common practice, gold standard and output keyphrases are stemmed to reduce the number of mismatches. One issue with the f-measure is that the ranks of the correct keyphrases are not taken into account. To evaluate the overall ranking performance of the models, we also report the Mean Average Precision (MAP) scores of the ranked lists of keyphrases. We use the Student’s paired t-test to assess statistical significance at the 0.05 level.

4.3 Replicability of results

In Table 3, we compare the results of our re-implementations against those reported in the original papers. We note that all models show comparable results. We observe the largest differences with original scores for CopyRNN (+2) and CorrRNN (−4.3) that can be easily explained by minor differences in training parameters.

Model	Dataset (metric)	Orig.	Ours
PositionRank	WWW (F@8)	12.3	11.7
MPRank	SemEval-2010 (F@10)	14.5	14.3
EmbedRank	Inspec (F@10)	37.1	35.6
CopyRNN	KP20k (F@10 on present)	26.2	28.2
CorrRNN	ACM (F@10 on present)	27.8	23.5

Table 3: Original vs. re-implementation scores.

4.4 Learning curves

The performance of neural keyphrase extraction models relies heavily on the availability of large amounts of labeled data. Yet, it is still unclear how much data is needed to achieve optimal performance. To provide insight into that question and investigate whether neural models would benefit from more training data, we trained CopyRNN using growing amounts (33%, 66% and 100%)⁷ of training data and show the corresponding learning curves in Figure 1. We see

⁶<https://github.com/epfml/sent2vec>

⁷Respectively the first 175 696; 351 393 and 527 090 samples for KP20k and the first 86 641; 173 282 and 259 923 samples for KPTimes.

that the model’s performance steadily improves as more training data is added, suggesting that available datasets may not be large enough. This is also suggested by recent work [48] where self-training (combining synthetic and gold samples for training) was shown to further improve performance. Moreover, adding more training data leads to larger gains on KPTimeS, whereas the performance of CopyRNN rapidly reaches a plateau on KP20K. The lack of annotation consistency in KP20k, that we highlighted in §2, may be the reason for this finding as learning a good model would be intuitively harder with non-uniformly labeled samples.

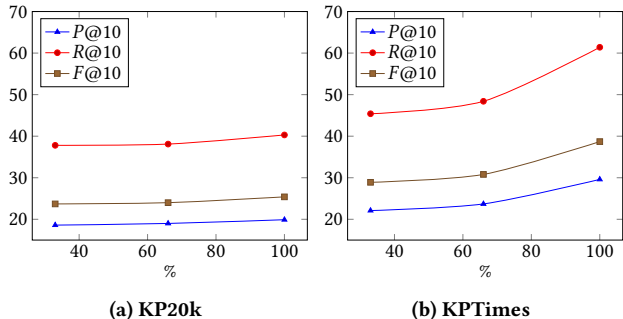


Figure 1: Performance of CopyRNN with different sizes of training data.

5 RESULTS

Results are presented in Table 2. First of all, we notice that no model significantly outperforms the baselines on all datasets. This is rather surprising, as one would expect that neural models would be consistently better than a simple TF×IDF model for example. Rather, we see that the TF×IDF baseline is very competitive on long documents, while the FirstPhrases baseline performs remarkably well, especially on news texts. Still, overall, CopyRNN achieves the best performance with, in the case of KPTimeS, MAP scores exceeding 50%. When we look at only unsupervised models, MPRank achieves the best results across datasets. Also, it comes as no surprise that Kea exhibits strong performance across datasets because it combines two effective features, as demonstrated by the results of the TF×IDF and FirstPhrases baselines. Conversely, despite the addition of mechanisms for promoting diversity in the output, CorrRNN is almost always outperformed by CopyRNN, suggesting that the added correlation constraints are not effective at filtering out spurious keyphrases.

In light of the above, we can now answer the following question: “How much progress have we made since early models?”. It is clear that neural-based models are the new state-of-the-art for keyphrase extraction, achieving F@10 scores up to three times that of previous models. That being said, CopyRNN, which is the best overall model, fails to consistently outperform the baselines on all datasets. One reason for that is the limited generalization ability of neural-based models [10, 15, 34], which means that their performance degrades on documents that differ from the ones encountered during training. This is besides confirmed by the extremely low performance of these models on DUC-2001 and KPCrowd. Much more work needs to be done in tackling this issue if neural models are to substitute for

older supervised models. Perhaps most disappointing is the fact that state-of-the-art unsupervised models are still challenged by the TF×IDF baseline. Here, we suspect the reasons are twofold. First, the models we have investigated do not use in-domain data which may not only limit their performance, but also, as in the case of EmbedRank that uses out-of-domain (Wikipedia) data, be detrimental to their performance. Second, unlike neural generative models, they are not able to produce keyphrases that do not occur in the source document, further limiting their potential effectiveness.

As outlined in §2, gold standards provided by lay annotators, such as authors and readers, exhibit strong inconsistency issues. One might therefore wonder “What is the impact of non-expert annotations on training and evaluating keyphrase extraction models?”. Intuitively, models evaluated against these annotations are likely to receive lower scores because they make training more difficult (that is, assigning different keyphrases to documents about the same topic may confuse the model) while increasing the number of false negatives during evaluation. This is exactly what we observe in Table 2 where the best scores for Inspec and KPTimeS, whose gold standards are provided by professional indexers, are higher in magnitude than those of the other datasets. Precisely quantifying how much impact lay annotations have on performance is no easy task as it implies a double-annotation process by both expert and non-expert annotators. Luckily enough, a small sample of documents from Inspec are also found in KP20k, allowing us to compare the performance of keyphrases models between both annotation types. Results are shown in Table 4. First, we see that overall performance is nearly cut in half when evaluating against author-provided gold standard, suggesting that reported scores in previous studies are arguably underestimated. Second, neural models again do not show their superiority against indexer-assigned keyphrases, which advocates the need for more experiments on datasets that include expert annotations.

Model	F@10		MAP	
	I	A	I	A
FirstPhrases	25.8	13.7	26.1	13.2
TextRank	33.4	12.2	29.6	9.3
TF×IDF	34.6	14.2	33.3	16.1
PositionRank	32.9	15.9	31.0	13.0
MPRank	26.4	13.8	27.6	13.6
EmbedRank	34.3	15.3	31.3	11.5
Kea	32.5	15.2	31.9	15.9
CopyRNN	33.7	28.9[‡]	29.8	33.8[‡]
CorrRNN	28.6	25.3	24.2	28.2
Avg.	31.3	17.2	29.4	17.2

Table 4: Results on a subset of 55 documents from Inspec for indexer (I) and author (A) gold annotations. [‡] indicates significance over every other model.

The third question we want to address in this study is “Which baselines and benchmark datasets should be included in future work for a better understanding of the pros and cons of a newly proposed model?”. Having strong baselines to compare with is of utmost

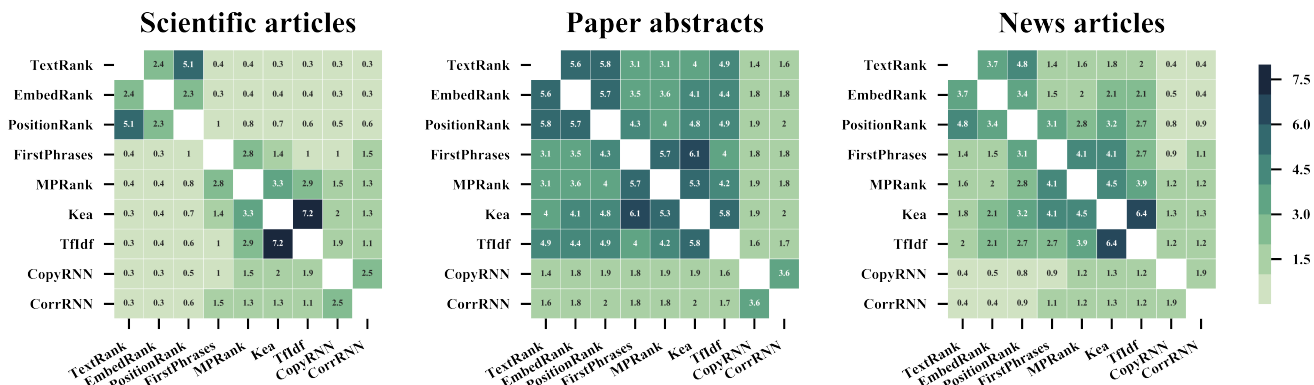


Figure 2: Average number of keyphrases in common between model outputs.

importance, and our results give an indication of which model is relevant. When properly trained, neural models drastically outperform all other models and represent the state-of-the-art. Since CopyRNN achieve the best results, it should be included in future work for comparison. In an unsupervised setting, or in a data-sparse scenario where neural models can not be applied, the picture is less clear. To help us understand which model is worth investigating, we conducted an additional set of experiments aimed at comparing the outputs from all models in a pairwise manner. The motivation behind these experiments is that including multiple models that behave similarly is of limited interest. Similarities between model outputs, viewed in terms of the number of keyphrases in common, are graphed as a heatmap in Figure 2. Overall, we observe different patterns for each source of documents. The shorter the document is, the more similar outputs are, which is mostly due to a smaller search space (that is, a smaller number of keyphrase candidates). We note that the three best unsupervised models, namely FirstPhrases, MPRank and TF×IDF, generate very similar keyphrases (up to 42% identical). Considering this, and given their reported performances (Table 2), we argue that TF×IDF (or KEA if seed training data is available) should be considered as strong unsupervised baseline in subsequent work.

These recommendations of baselines also affect the choice of which benchmark datasets one has to use. As neural models are data-hungry, KP20k and KPTimes are the default options for paper abstracts and news articles. For scientific articles, we recommend using SemEval for two reasons: 1) it is widely used by existing studies; and 2) it provides a double-annotated gold standard (author- and reader-assigned keyphrases) that alleviates annotation inconsistencies to some extent.

Our experiments highlight several issues in evaluating keyphrase extraction models with existing benchmark datasets. Another way of assessing the effectiveness of these models would be to explore their impact on other tasks as an extrinsic evaluation. To the best of our knowledge, there is no previously published research on that matter despite many downstream tasks that already benefit from keyphrase information such as article recommendation [12] or browsing interfaces [20] in digital libraries. This points to an

interesting future direction that allows for a deeper understanding of the limitations of current models.

6 CONCLUSION

This paper presents a large scale evaluation of keyphrase extraction models conducted on multiple benchmark datasets from different sources and domains. Results indicate that keyphrase extraction is still an open research question, with state-of-the-art neural-based models still challenged by simple baselines on some datasets. We hope that this work will serve as a point of departure for more rigorous analysis and evaluation of proposed keyphrase extraction models. We provide all the code and data on a public repository⁸, as well as a public leaderboard to facilitate the comparison between models.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments. This work was supported by the French National Research Agency (ANR) through the DELICES project (ANR-19-CE38-0005-01).

REFERENCES

- [1] Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. SemEval 2017 Task 10: ScienceIE - Extracting Keyphrases and Relations from Scientific Publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, 546–555. <http://www.aclweb.org/anthology/S17-2091>
- [2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
- [3] Frances H Barker, Douglas C Veal, and Barry K Wyatt. 1972. Comparative efficiency of searching titles, abstracts, and index terms in a free-text data base. *Journal of Documentation* 28, 1 (1972), 22–36.
- [4] Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple Unsupervised Keyphrase Extraction using Sentence Embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Brussels, Belgium, 221–229. <http://www.aclweb.org/anthology/K18-1022>
- [5] Gábor Berend. 2011. Opinion Expression Mining by Exploiting Keyphrase Extraction. In *Proceedings of 5th International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, Chiang Mai, Thailand, 1162–1170. <http://www.aclweb.org/anthology/I11-1130>

⁸https://github.com/ygorg/JCDL_2020_KPE_Eval

- [6] Florian Boudin. 2016. pke: an open source python-based keyphrase extraction toolkit. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*. The COLING 2016 Organizing Committee, Osaka, Japan, 69–73. <http://aclweb.org/anthology/C16-2015>
- [7] Florian Boudin. 2018. Unsupervised Keyphrase Extraction with Multipartite Graphs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 667–672. <http://www.aclweb.org/anthology/N18-2105>
- [8] Florian Boudin, Hugo Mougard, and Damien Cram. 2016. How Document Pre-processing affects Keyphrase Extraction Performance. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*. The COLING 2016 Organizing Committee, Osaka, Japan, 121–128. <http://aclweb.org/anthology/W16-3917>
- [9] Cornelia Caragea, Florin Adrian Bulgarov, Andreea Godea, and Sujatha Das Gollapalli. 2014. Citation-Enhanced Keyphrase Extraction from Research Papers: A Supervised Approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1435–1446. <http://www.aclweb.org/anthology/D14-1150>
- [10] Jun Chen, Xiaoming Zhang, Yu Wu, Zhao Yan, and Zhoujun Li. 2018. Keyphrase Generation with Correlation Constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 4057–4066. <http://www.aclweb.org/anthology/D18-1439>
- [11] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1724–1734. <http://www.aclweb.org/anthology/D14-1179>
- [12] Andrew Collins and Jöran Beel. 2019. Document Embeddings vs. Keyphrases vs. Terms for Recommender Systems: A Large-Scale Online Evaluation. In *19th ACM/IEEE Joint Conference on Digital Libraries, JCDL 2019, Champaign, IL, USA, June 2-6, 2019*. 130–133. <https://doi.org/10.1109/JCDL.2019.00027>
- [13] David A. Evans and Chengxiang Zhai. 1996. Noun Phrase Analysis in Large Unrestricted Text for Information Retrieval. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Santa Cruz, California, USA, 17–24. <https://doi.org/10.3115/981863.981866>
- [14] Corina Florescu and Cornelia Caragea. 2017. PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1105–1115. <http://aclweb.org/anthology/P17-1102>
- [15] Ygor Gallina, Florian Boudin, and Beatrice Daille. 2019. KPTime: A Large-Scale Dataset for Keyphrase Generation on News Documents. In *Proceedings of the 12th International Conference on Natural Language Generation*. Association for Computational Linguistics, Tokyo, Japan, 130–135. <https://doi.org/10.18653/v1/W19-8617>
- [16] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*. Association for Computational Linguistics, Melbourne, Australia, 1–6. <https://doi.org/10.18653/v1/W18-2501>
- [17] Jade Goldstein and Jaime Carbonell. 1998. SUMMARIZATION: (1) USING MMR FOR DIVERSITY-BASED RERANKING AND (2) EVALUATING SUMMARIES. In *Proceedings of the TIPSTER Text Program: Phase III*. Association for Computational Linguistics, Baltimore, Maryland, USA, 181–195. <https://doi.org/10.3115/1119089.1119120>
- [18] Sujatha Das Gollapalli, Cornelia Caragea, Xiaoli Li, and C. Lee Giles (Eds.). 2015. *Proceedings of the ACL 2015 Workshop on Novel Computational Approaches to Keyphrase Extraction*. Association for Computational Linguistics, Beijing, China. <http://www.aclweb.org/anthology/W15-36>
- [19] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 1631–1640. <http://www.aclweb.org/anthology/P16-1154>
- [20] Carl Gutwin, Gordon Paynter, Ian Witten, Craig Nevill-Manning, and Eibe Frank. 1999. Improving Browsing in Digital Libraries with Keyphrase Indexes. *Decis. Support Syst.* 27, 1-2 (Nov. 1999), 81–104. [https://doi.org/10.1016/S0167-9236\(99\)00038-X](https://doi.org/10.1016/S0167-9236(99)00038-X)
- [21] Kazi Saidul Hasan and Vincent Ng. 2014. Automatic Keyphrase Extraction: A Survey of the State of the Art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 1262–1273. <http://www.aclweb.org/anthology/P14-1119>
- [22] Anette Hulth. 2003. Improved Automatic Keyword Extraction Given More Linguistic Knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, Michael Collins and Mark Steedman (Eds.). 216–223. <http://www.aclweb.org/anthology/W03-1028.pdf>
- [23] Anette Hulth and Beáta B. Megyesi. 2006. A Study on Automatically Extracted Keywords in Text Categorization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Sydney, Australia, 537–544. <https://doi.org/10.3115/1220175.1220243>
- [24] Richard Hussey, Shirley Williams, Richard Mitchell, and Ian Field. 2012. A comparison of automated keyphrase extraction techniques and of automatic evaluation vs. human evaluation. *International Journal on Advances in Life Sciences* 4, 3 and 4 (2012), 136–153. <http://centaur.reading.ac.uk/32266/>
- [25] Chris Kedzie, Kathleen McKeown, and Hal Daume III. 2018. Content Selection in Deep Learning Models of Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 1818–1828. <http://www.aclweb.org/anthology/D18-1208>
- [26] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. SemEval-2010 Task 5 : Automatic Keyphrase Extraction from Scientific Articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Uppsala, Sweden, 21–26. <http://www.aclweb.org/anthology/S10-1004>
- [27] Mikalai Krapivin, Aliaksandr Autaeu, and Maurizio Marchese. 2009. *Large dataset for keyphrases extraction*. Technical Report. University of Trento.
- [28] Marina Litvak and Mark Last. 2008. Graph-Based Keyword Extraction for Single-Document Summarization. In *Coling 2008: Proceedings of the workshop Multi-source Multilingual Information Extraction and Summarization*. Coling 2008 Organizing Committee, Manchester, UK, 17–24. <http://www.aclweb.org/anthology/W08-1404>
- [29] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 1412–1421. <http://aclweb.org/anthology/D15-1166>
- [30] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. 55–60. <http://www.aclweb.org/anthology/P/P14/P14-5010>
- [31] Yuqing Mao and Zhiyong Lu. 2017. MeSH Now: automatic MeSH indexing at PubMed scale via learning to rank. *Journal of Biomedical Semantics* 8, 1 (17 Apr 2017), 15. <https://doi.org/10.1186/s13326-017-0123-3>
- [32] Daniel Marcu. 1997. The Rhetorical Parsing of Unrestricted Natural Language Texts. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Madrid, Spain, 96–103. <https://doi.org/10.3115/976909.979630>
- [33] Luis Marujo, Anatole Gershman, Jaime Carbonell, Robert Frederking, and João P. Neto. 2012. Supervised Topical Key Phrase Extraction of News Stories using Crowdsourcing, Light Filtering and Co-reference Normalization. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12) (23-25)*, Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (Eds.). European Language Resources Association (ELRA), Istanbul, Turkey.
- [34] Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep Keyphrase Generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Vancouver, Canada). Association for Computational Linguistics, 582–592. <https://doi.org/10.18653/v1/P17-1054>
- [35] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Texts. In *Proceedings of EMNLP 2004*, Dekang Lin and Dekai Wu (Eds.). Association for Computational Linguistics, Barcelona, Spain, 404–411.
- [36] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS 2017 Workshop Autodiff*.
- [37] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 24, 5 (1988), 513–523. [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0)
- [38] Alexander Thorsten Schutz. 2008. Keyphrase extraction from single documents in the open domain exploiting linguistic and statistical methods. *Master's thesis, National University of Ireland* (2008).
- [39] Sanjay Sood, Sara Owsley, Kristian J. Hammond, and Larry Birnbaum. 2007. TagAssist: Automatic Tag Suggestion for Blog Posts. In *Proceedings of the First International Conference on Weblogs and Social Media, ICWSM 2007, Boulder, Colorado, USA, March 26-28, 2007*. <http://www.icwsm.org/papers/paper10.html>
- [40] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing*

- Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3104–3112. <http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf>
- [41] Nedelina Teneva and Weiwei Cheng. 2017. Saliency Rank: Efficient Keyphrase Extraction with Topic Modeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Vancouver, Canada, 530–535. <http://aclweb.org/anthology/P17-2084>
- [42] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling Coverage for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 76–85. <https://doi.org/10.18653/v1/P16-1008>
- [43] Xiaojun Wan and Jianguo Xiao. 2008. Single Document Keyphrase Extraction Using Neighborhood Knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2 (Chicago, Illinois) (AAAI'08)*. AAAI Press, 855–860. <http://dl.acm.org/citation.cfm?id=1620163.1620205>
- [44] Rui Wang, Wei Liu, and Chris McDonald. 2014. How Preprocessing Affects Unsupervised Keyphrase Extraction. In *Computational Linguistics and Intelligent Text Processing*, Alexander Gelbukh (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 163–176.
- [45] Rui Wang, Wei Liu, and Chris McDonald. 2015. Using Word Embeddings to Enhance Keyword Identification for Scientific Publications. In *Databases Theory and Applications*, Mohamed A. Sharaf, Muhammad Aamir Cheema, and Jianzhong Qi (Eds.). Springer International Publishing, Cham, 257–268.
- [46] Ian H Witten, David Bainbridge, and David M Nichols. 2009. *How to build a digital library*. Morgan Kaufmann.
- [47] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. KEA: Practical Automatic Keyphrase Extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries (Berkeley, California, USA) (DL'99)*. ACM, New York, NY, USA, 254–255. <https://doi.org/10.1145/313238.313437>
- [48] Hai Ye and Lu Wang. 2018. Semi-Supervised Learning for Neural Keyphrase Generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 4142–4153. <http://www.aclweb.org/anthology/D18-1447>
- [49] Torsten Zesch and Iryna Gurevych. 2009. Approximate Matching for Evaluating Keyphrase Extraction. In *Proceedings of the International Conference RANLP-2009*. Association for Computational Linguistics, Borovets, Bulgaria, 484–489. <http://www.aclweb.org/anthology/R09-1086>
- [50] Chengxiang Zhai. 1997. Fast Statistical Parsing of Noun Phrases for Document Indexing. In *Fifth Conference on Applied Natural Language Processing*. Association for Computational Linguistics, Washington, DC, USA, 312–319. <https://doi.org/10.3115/974557.974603>