

Exploring Models for Automatic Keyword Labelling of Scientific Documents

Jorge Gabín^{1,2,*†}, M. Eduardo Ares^{1†} and Javier Parapar^{2†}

¹Linknovate Science, Rúa das Flores 33, Roxos, Santiago de Compostela, A Coruña, 15896, Spain

²IRLab, CITIC, Computer Science Department, University of A Coruña, A Coruña, 15071, Spain

Abstract

Automatic keyword labelling methods generate a set of short phrases for a given document providing a short and good description of its content. Those labels are critical in tasks such as exploratory search and for improving the information discovery experience. This paper presents a novel keyword labelling model based on text-to-text transfer transformers (T5). We train a T5 model to generate keywords from academic documents content. We name this model docT5keywords. We compare our proposal with the state-of-the-art EmbedRank model, based on Sent2Vec embeddings and even with the keywords manually assigned by the author for representing their writings.

Our proposal does not merely extract fragments of the texts but also may produce unseen labels. We commonly refer to these models as creative models. Classical evaluation based on matching against a set of golden truth labels extracted from the texts is not the best alternative when examining the performance of creative methods. Therefore, we also present an alternative user-based evaluation methodology for creative keyword generation models. In our user study, we examine the performance of the tested models using four expert assessors while analysing the assessor agreement and the correlation with the classical offline evaluation methodologies.

Keywords

Keyword Labelling, Keyword Generation, Text-To-Text Transfer Transformers, User-based Evaluation

1. Introduction

Having documents with suitable keywords or labels is crucial for exploratory search [1] and also for improving the user experience [2] during the discovery task. In addition, keyphrases have many useful applications such as enabling semantic and faceted search [3, 4], query expansion [5] or document clustering [6] and classification [7].

Unfortunately, despite the known importance of these short descriptions for the documents, most of the documents indexed in search engines either miss those keywords or have low-quality ones. Therefore, keyword extraction and generation models are needed for filling that gap. Keyword extraction and generation techniques use documents' content to extract or generate keywords representing them. Recent advances in the NLP (Natural Language Processing) field

CIRCLE'22: Joint Conference of the Information Retrieval Communities in Europe, July 04–07, 2022, Toulouse, France

*Corresponding author.


†These authors contributed equally.

✉ jorge@linknovate.com (J. Gabín); eduardo@linknovate.com (M. E. Ares); javier.parapar@udc.es (J. Parapar)

🌐 <https://www.dc.fi.udc.es/~parapar> (J. Parapar)

🆔 0000-0002-5494-0765 (J. Gabín); 0000-0003-3807-5692 (M. E. Ares); 0000-0002-5997-8252 (J. Parapar)

© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

allowed the appearance of unsupervised and supervised models that achieve good results on some keyword labelling tasks. Despite this, some limitations exist on the models' capacity to produce unseen labels as most of them merely extract phrases or words from the given text to obtain the keywords.

Keyword labelling of scientific documents is essential for carrying out critical tasks in the scientific domain, like article recommendation [8], potential reviewers' identification and analysis of contents trends. Moreover, in the case of academic publications, it is common to ask authors to label their writings manually. Therefore, researchers tend to train and test their keyword extraction models in this domain, given the availability of human-produced references. Transfer learning [9], in which a model is first trained on a data-intensive task before being refined on a subsequent task, is a de-facto standard in NLP. The effectiveness of transfer learning has given rise to a diversity of approaches, methodologies and practices in this area. Raffel et al. [10] recently presented T5, a new successful model for transfer learning for NLP. Text-to-Text Transfer Transformer (T5) is a unified framework that converts all text-based language problems into a text-to-text format rather than following a multi-step approach.

This paper presents `docT5keywords` - a novel T5-based method that automatically generates keywords from documents. This new technique is simple as it only requires documents' abstracts to produce the keywords related to them. Also, as it is a text-to-text model, `docT5keywords` does not need to pull phrases or words from the given text. Instead, it generates keywords that may or may not appear in the context, which addresses the lack of creativity issue of older keyphrase extraction models.

To evaluate our models' performance, we compare it against a state-of-the-art unsupervised keyphrase extraction model based on sentence embeddings, `EmbedRank` [11]. This model achieved great results on several keyword extraction datasets without neglecting performance or diversity when extracting keywords at inference time.

We carry out the evaluation in `INSPEC` [12] and `NUS` [13], two classical keyphrase extraction datasets compound of scientific publications, to compare the models in a standard and reproducible way. Due to the limitations of this kind of dataset on evaluating creative models (like `docT5keywords`), we include an alternative user-based evaluation methodology for correctly assessing creative keyword generation models. For this evaluation we use other two datasets: `MAG` [14], for training the model and `CDS 2016`, for testing. We will also use these two datasets on the offline evaluation to address a better comparison.

This study presents the performance of the tested models against each other and the keywords assigned by the authors. Furthermore, we analyse the evaluation made by four assessors, the assessor agreement [15], and the correlation with the classical offline evaluation methodologies. The main contributions of this work are:

- A novel keyword generation model based on T5 that automatically generates keywords using only documents abstracts. This new model performs great at inference time (necessary for its use in production systems) and produces creative results.
- A comparison of our model against a state-of-the-art unsupervised keyword extraction model (`EmbedRank`), presenting both the results over classical evaluation datasets and a user study where four assessors evaluate the quality of the labels.

2. Related work

This section briefly overviews the existing keyphrase extraction methods, deepening the EmbedRank model. We also introduce the transfer learning basis that originated the T5 model that we built our proposal upon.

Previous works on keyword extraction have analyzed the pros and limitations of these techniques for describing documents. The reader may find that Papagiannopoulou and Tsoumakas [16] present an excellent and recent survey on keyphrase extraction. This work shows a thorough comparison between a large set of unsupervised and supervised keyphrase extraction models. They found limitations on existing evaluation methodologies. In particular, they explore how classical exact matching evaluation differs from recent partial-matching proposals [17]. They show how one significant shortcoming of exact matching evaluation is that it penalizes methods even if they anticipate semantically comparable keywords to the golden ones. However, they also found that the alternative partial matching approach rewards algorithms that predict terms that exist in golden keyphrases, even if the predicted keyphrases are not appropriate for the accompanying article.

2.1. EmbedRank

EmbedRank [11] is an unsupervised keyphrase extraction method based on embeddings. It mainly follows these three steps: first, candidate phrases are extracted from the text, only keeping phrases that consist of zero or more adjectives followed by at least one noun; second, both the document and the phrases are represented as embeddings (using the same high-dimensional vector space); finally, phrases are ranked to select the output keywords.

EmbedRank uses embeddings capabilities to capture semantic relatedness of words, phrases and documents to rank the candidate terms extracted from the text. Thus, EmbedRank computes the document embedding and each candidate phrase's embedding using the same algorithm in its second phase. Using the embeddings built in the previous phase, in the third and last phase, EmbedRank selects the top keywords according to the cosine distance to the document they belong to.

One key aspect of this model is the possibility of changing the embedding model to the one which best fits our necessity. Both Doc2Vec [18] and Sent2Vec [19] alternatives are tested and compared on [11]. Sent2Vec turned out to be much faster at inference time (something crucial for production environments). Also, when working with short and medium-length documents, the model based on Sent2Vec outperforms previous state-of-the-art unsupervised models.

Along with EmbedRank, authors present EmbedRank++, an alternative model which uses Maximal Marginal Relevance (MMR) [20] to increase keyphrase diversity. EmbedRank base model only considers phrase informativeness which leads to redundant keyphrases. This feature negatively impacts users' experience in scenarios where they directly see and use these keyphrases. Moreover, this problem intensifies when extracting the top N keywords, where, given the limited number of keywords presented to the user, having near duplicates or redundant variations of keywords is a waste of space. EmbedRank++ aims to address this issue by adapting MMR to the keyphrase extraction task combining keywords informativeness and diversity. It includes a hyperparameter to control the trade-off between informativeness and diversity,

making it possible to adapt the model to each situation.

In conclusion, this fully unsupervised model based on embeddings is an excellent alternative for keyword extraction in a wide variety of domains.

2.2. Transfer learning

Many old machine learning methods only work well, relying on a common assumption: obtained training and test data must belong to the same feature space and distribution. Nevertheless, there are many real-world domains or applications where it is expensive or impossible to retrieve the training data needed to build the models. Semisupervised machine learning methods [21] address those situations using only a small amount of labelled data but leveraging a larger number of unlabelled examples. However, these techniques still assume that the labelled and unlabelled data distribution is the same.

On the other side, transfer learning [9] allows domains, tasks and distributions used in training and test phases to be different. These methods aim to improve learners' performance on target domains by transferring the knowledge in different but related source domains.

We can categorize transfer learning techniques into three classes: inductive, transductive and unsupervised transfer learning. Both inductive and transductive methods are supervised approaches. In the inductive setting, the target task is different from the source task, while the domain can either be the same or not. In this case, we usually need labelled data in the target domain.

Meanwhile, in the transductive case, both source and target tasks must be the same, but the source and target domain are different. In this situation, many labelled data is available in the source domain, while none is available in the target domain.

Finally, under the unsupervised case, we do not have any labelled data available in the source or target domain. Moreover, the target task is different but related to the source task, as it happened in the inductive transfer learning method.

Many recent state-of-the-art NLP models rely on these learning techniques, i.e. they are pre-trained models on large datasets that we later fine-tune for specific tasks (e.g. BERT [22] or T5 [10]).

2.3. Text-to-Text Transfer Transformer (T5)

Most NLP tasks require machine learning models to develop general-purpose knowledge so that the model can “comprehend” the text. However, recent approaches do not explicitly train models on this general task. Instead, these models learn this general-purpose knowledge via an auxiliary task.

More recent approaches use transfer learning techniques to acquire this knowledge. As we explained in the previous section, on these approaches, models are first pre-trained on a data-rich task and then fine-tuned on specific downstream tasks. *Transferring* knowledge from the first to the latter task.

While in other fields such as computer vision, models commonly use supervised transfer learning, modern state-of-the-art NLP favour unsupervised transfer learning.

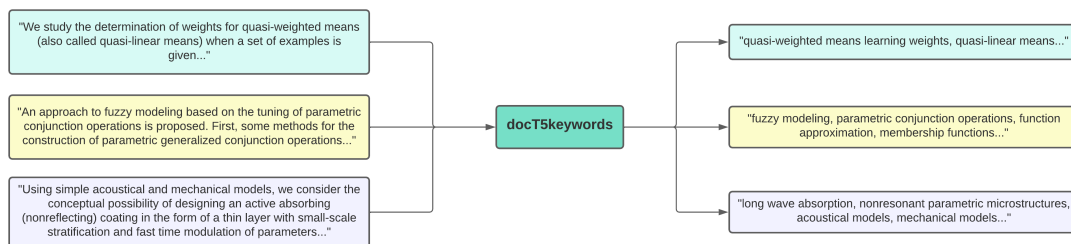


Figure 1: A diagram of our text-to-text framework for the downstream task of keyword generation. (Examples taken from *Inspec* dataset)

The main idea behind T5 [10] is to treat every text processing problem as a text-to-text problem. The T5 model is a slight adaptation of the original Transformer [23]. It removes the Layer Form bias, placing the layer normalization [24] outside the residual path and using a different embedding position scheme. Instead of using a fixed embedding for each position, it uses relative position embeddings, a new, more common alternative. The architecture learns the embeddings according to the offset between the key and query being compared in the self-attention mechanism. Note that an attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. Along with T5, authors also leverage the "Colossal Clean Crawled Corpus" (C4), a cleaned version of the web extracted text from April 2019 by Common Crawl. T5 models are pre-trained using this dataset. Because T5 aims to treat every text processing problem as a text-to-text problem, it was necessary to prove its performance on many downstream tasks. Experiments on [10] and recent work in NLP have shown the excellent performance achieved by this model on a wide variety of tasks.

3. Proposal

Our proposal presents an unexplored keyword generation model that takes advantage of the text generation capacities of the novel T5 text-to-text architecture. We name this model `docT5keywords`.

We train the model as follows: for each document, the downstream task is to deliver a set of keywords that properly represent that document based on its content. More specifically, we train the model only using the abstract of academic publications. Figure 1 shows some examples of how the model works.

In the training phase, given a set of pairs (document abstract, document keywords), we feed as input the abstract and fine-tune T5 to generate keywords that suit the document. In the inference stage, we will need to feed our fine-tuned model with document abstracts, and it will output keywords for each of them.

Using documents' abstracts and keywords from three scientific papers collections, we built models for three different experiments, as we explain in Section 4.

The main advantage of using a text-to-text model like T5 for a document keyword labelling task is its possibility of generating keywords that are not in the input text, giving the chance of

building creative models.

In contrast, most models used to deal with this task are based on keyword extraction methods that first retrieve words and phrases from the document and then process them to select the output keywords, limiting models' creativity.

Even though models' creativity is a crucial aspect of the keyword labelling of documents task, creative models have a significant drawback. Classical evaluation datasets on keyword extraction tasks do not consider this characteristic of models, as they almost always label documents with keywords that appear in the given text.

Given the above, we present, alongside our model, an alternative user-based evaluation methodology comparing docT5keywords, EmbedRank and author keywords. This study will gather the evaluations of four expert assessors to assess our creative model fairly and compare its performance against EmbedRank's and author's keywords.

4. Experimental setup

In this section, we describe the datasets, the evaluation methods and the parameters used to assess the performance of our model.

4.1. Datasets

4.1.1. Inspec

This dataset [12] is a collection of 2,000 titles and abstracts from scientific journal papers. Each document from this collection has two sets of keyphrases assigned by the indexers: the controlled keyphrases, which appear in the Inspec thesaurus, and the uncontrolled keyphrases, which do not necessarily appear in the thesaurus. For our experiments, we will only use the set of uncontrolled keyphrases

4.1.2. National University of Singapore (NUS)

NUS corpus for keyword extraction [13] contains 211 long full scientific conference papers with a length between 4 and 12 pages. Each document provides several keywords: one created by the authors and, potentially, several others created by annotators. Following the job done on [11], we evaluate using the union of all sets of keywords.

4.1.3. Microsoft Academic Graph (MAG)

MAG [14] is a heterogeneous graph containing scientific publication records, citation relationships between those publications, as well as authors, institutions, journals, conferences, and fields of study.

We used MAG papers published in the Open Academic Graph v1 (OAG v1), a dataset generated by linking two large academic graphs: Microsoft Academic Graph (MAG) and AMiner. The decision to use the first version of OAG instead of the second one is because MAG papers included in OAG v2 did not contain author keywords in their documents.

This version of OAG includes 166,192,182 papers from MAG. We preprocessed these documents by deleting undesired characters and only keeping those that contained both abstract and keywords.

4.1.4. TREC - Clinical Decision Support (CDS) 2016

Following the work done in previous years (2014 and 2015), in 2016, TREC launched the third edition of the Clinical Decision Support (CDS) task.

The goal of this task relies on retrieving biomedical articles relevant for answering generic clinical questions about medical records. In contrast with previous years, actual electronic health record (EHR) patient records were used instead of synthetic cases.

The document collection used for the task is the Open Access Subset of PubMed Central (PMC). PMC is an online digital database of freely available full-text biomedical literature. This database is constantly updated, so they used a snapshot from March 28, 2016, containing 1.25 million articles.

We filtered the collection by keeping only articles with abstract and author keywords (needed for the user-based evaluation and used as the ground truth in the respective offline experiment). Then, we used documents' abstracts to infer new keywords.

4.2. Offline evaluation

We first compare docT5keywords to EmbedRank using two classic keyword extraction datasets (Inspec and NUS). After that, we also compare models' performance on the dataset used for TREC CDS-2016.

For the first collection (Inspec), we used the training and validation subsets (totalling 1500 documents) to fine-tune T5. For testing, we used the provided subset (500 documents) without any modification for both our model and EmbedRank. We used the title and abstract as context and keywords as labels for each record.

In the case of the NUS dataset, we had to split the collection ourselves as no previous split existed. Therefore, we decided to keep 150 documents for training and 59 for testing (we discarded two records as they did not have an abstract). Furthermore, it was necessary to limit the size of the documents' abstracts for both training and test phases because our implementation for T5 only supports up to 512 tokens as context. However, we did not apply this length limitation to EmbedRank's test split as it is a particular feature of our model which may lead to worse results in models like EmbedRank.

Finally, we decided to include CDS in our offline evaluation to compare offline and user-based studies' results. In contrast with the previous datasets, we did not use a split of the CDS collection to train our model. Instead, we used a two million random sample of documents from the MAG dataset for training and a total of 250 papers (same subset for online and offline assessment) from the CDS collection for testing. Of course, this is a more challenging scenario, as the topics from the articles from the training and test splits may differ, but we wanted to see how this shift may affect the relative performance of the model. We only used abstracts to train and infer docT5keywords in MAG and CDS documents. These abstracts were preprocessed not to exceed the maximum tokens length of 512.

For the EmbedRank method, we tried different configurations regarding the trade-off between informativeness and diversity (λ). We evaluated the model on its full informativeness and diversity versions. We also assessed its default option ($\lambda = 0.55$) set in the code leveraged by the authors¹. Regarding the embedding model used for EmbedRank, we choose Sent2Vec as authors reported to perform better than Doc2Vec in overall performance.

4.2.1. Offline metrics

In terms of evaluation metrics, we calculated precision (P), recall (R) and F_1 . We compared models' generated keywords against datasets' ground truth.

We compute those metrics following the exact match evaluation approach, where the number of correctly matched phrases with the golden ones are determined based on string matching. As we commented in Section 2 (Related Work), the exact matching approach is considered suboptimal as it penalizes methods even if they find semantically equivalent keywords to the ones in the golden set. Moreover, we cannot directly compare the results with the references because the models may generate keywords with slight differences from the ground truth, such as number or verb tense. To alleviate that problem of exact matching evaluation, we decided to process both models' outputs and datasets' reference keywords using a stemmer and then deleting characters like dashes or even spaces, as some words can be spelt in different ways. Table 1 shows the results of the offline evaluation which we will explain and discuss in Section 5.

4.3. User-based evaluation

As previously commented, classical evaluation datasets are not the best option to assess creative models' performance. The main problem regarding the keyword generation task is that these datasets cannot identify if keywords not in the ground truth set are suitable for the document. Therefore, we propose a user-based evaluation experiment as an alternative to the previously presented offline evaluation.

For the online experiment, following the approach of the offline evaluation, we used a subset of two million documents from the MAG collection to train our model and a split of 250 papers from the CDS-2016 to evaluate them with the following strategy. Four expert assessors carried out our user study, and each one assessed the keywords of 100 CDS records; 50 of them were shared by the four experts, whereas the other 50 were different. The rationale behind having a split of documents in the intersection of all the assessments was for studying assessors' agreement while rating keywords.

For each document of the test set, we provided assessors with its title and abstract and a maximum of five keywords of each type (generated by docT5keywords, extracted by EmbedRank and annotated by authors). Figure 2 shows the interface we provided to the assessors to evaluate the keywords. Assessors had to score each keyword between 0 and 3, with 0 being a non-suitable keyword and 3 a perfect keyword for the document. The matching between each value and its meaning goes as follows: inappropriate (0), somewhat related (1), reasonable (2) and very good (3).

¹<https://github.com/swisscom/ai-research-keyphrase-extraction>

Automatic Keywords Generation Evaluation

Isolation, growth and identification of colony-forming cells with erythroid, myeloid, dendritic cell and NK-cell potential from human fetal liver

Abstract

The study of hematopoietic stem cells (HSCs) and the process by which they differentiate into committed progenitors has been hampered by the lack of in vitro clonal assays that can support erythroid, myeloid and lymphoid differentiation. We describe a method for the isolation from human fetal liver of highly purified candidate HSCs and progenitors based on the phenotypes CD38 - CD34 ++ and CD38 + CD34 ++, respectively. We also describe a method for the growth of colony-forming cells (CFCs) from these cell populations, under defined culture conditions, that supports the differentiation of erythroid, CD14/CD15 + myeloid, CD1a + dendritic cell and CD56 + NK cell lineages. Flow cytometric analyses of individual colonies demonstrate that CFCs with erythroid, myeloid and lymphoid potential are distributed among both the CD38 - and CD38 + populations of CD34 ++ progenitors.

Generated Keywords

lymphoid differentiation

Inappropriate Somewhat related Reasonable Very good

phenotypes cd38

Inappropriate Somewhat related Reasonable Very good

cell differentiation

Inappropriate Somewhat related Reasonable Very good



SUBMIT

Figure 2: User-based evaluation web interface. *Note: some of the keywords were removed due to space constraints.*

We calculated each model’s mean performance based on the assessors’ scores and the assessors’ agreement on the evaluation of the shared documents. Also, the nature of this kind of evaluation allows us to assess the creative part of our model. Therefore, we also computed our model’s mean score using only creative keywords (i.e. keywords not present in the text used for inference) and not creative ones. Finally, we computed Cohen’s Kappa [25] and Fleiss’ Kappa [26] scores to evaluate the rate of concordance between our assessors.

4.4. Experimental settings

Closing this section, we will show the parameters used in the learning and inferring phase when using our model for each dataset.

Starting from the T5-base model, we followed almost the same approximation on all three datasets regarding the training phase. Batch size, learning rate, and maximum input and output tokens were the same for all of them, with the following values: the batch size of 256, a learning rate of 10^{-3} , maximum input tokens of 512 and maximum output tokens of 64.

The parameter we had to vary depending on the dataset was the number of epochs. We followed an early stopping approximation for Inspec and NUS datasets, training the model until the loss was stable, with 80 epochs for the first and 64 for the second.

On the other hand, as the size of the training set was significantly bigger, we followed a different strategy for the MAG dataset. Instead of waiting until the loss was stable, we trained our model

Table 1

Comparison between our method and EmbedRank variations on the three datasets. Precision (P), Recall (R) and F-score (F_1) at 5, 10 and 15 are reported.

| N | Method | Inspec | | | NUS | | | CDS | | |
|----|--------------------------------------|--------------|--------------|--------------|--------------|-------------|--------------|--------------|--------------|--------------|
| | | P | R | F_1 | P | R | F_1 | P | R | F_1 |
| 5 | EmbedRank s2v | 40.24 | 20.05 | 27.14 | 18.64 | 3.95 | 6.51 | 11.36 | 10.97 | 11.16 |
| | EmbedRank++ s2v ($\lambda = 0.55$) | 40.08 | 20.40 | 27.03 | 13.90 | 2.94 | 4.85 | 8.32 | 8.03 | 8.17 |
| | EmbedRank++ s2v ($\lambda = 0.0$) | 18.96 | 9.65 | 12.79 | 4.75 | 1.04 | 1.66 | 5.12 | 4.94 | 5.03 |
| | docT5keywords | 56.34 | 27.58 | 37.03 | 20.89 | 4.38 | 7.24 | 8.95 | 7.50 | 8.15 |
| 10 | EmbedRank s2v | 35.40 | 35.60 | 35.50 | 16.44 | 6.97 | 9.78 | 10.05 | 19.38 | 13.23 |
| | EmbedRank++ s2v ($\lambda = 0.55$) | 33.02 | 33.22 | 33.12 | 13.05 | 5.23 | 7.76 | 6.84 | 13.20 | 9.02 |
| | EmbedRank++ s2v ($\lambda = 0.0$) | 18.82 | 18.93 | 18.88 | 5.76 | 2.44 | 3.43 | 5.08 | 9.81 | 6.70 |
| | docT5keywords | 52.73 | 43.11 | 47.44 | 16.85 | 6.74 | 9.63 | 7.97 | 10.81 | 9.17 |
| 15 | EmbedRank s2v | 31.98 | 46.49 | 37.90 | 13.52 | 8.54 | 10.47 | 8.82 | 25.41 | 13.09 |
| | EmbedRank++ s2v ($\lambda = 0.55$) | 28.89 | 41.99 | 34.23 | 10.01 | 6.69 | 8.44 | 6.43 | 18.53 | 9.55 |
| | EmbedRank++ s2v ($\lambda = 0.0$) | 20.10 | 29.21 | 23.81 | 6.82 | 4.30 | 5.28 | 4.69 | 13.51 | 6.70 |
| | docT5keywords | 50.88 | 45.86 | 48.24 | 16.25 | 8.32 | 11.00 | 7.49 | 12.51 | 9.37 |

for around two days, equivalent to training the model for two epochs.

Finally, we used the same parameters for all datasets concerning the inference phase, with the same batch size and maximum input and output tokens as the training step.

Concerning EmbedRank, we used its enhanced version, which uses MMR to provide more diverse results, setting the trade-off between informativeness and diversity to 0.55. We also decided to use Sent2Vec as the embedding model.

5. Results and discussion

This section reports how docT5keywords performs against EmbedRank when using traditional exact matching offline evaluation on classic collections together with the results of our user study.

5.1. Offline evaluation

As shown in Table 1, docT5keywords outperforms EmbedRank on two out of the three datasets in terms of precision, recall and F_1 score. Note that we reproduced the results of EmbedRank on both Inspec and NUS datasets which led to slight variances from the results reported on [11]. Concerning EmbedRank, we can see that increasing models' diversity leads to worse results on every dataset. This fact supports our idea that classic datasets are not yet prepared to correctly evaluate models where diversity and creativity are involved. On the other hand, we spotted significant better values of the EmbedRank model on the NUS dataset than the ones originally reported [11]. These improvements may be related to the choice of the fragment of the document used for inferring the keywords (we only used the abstract of the articles).

Table 2

Comparison between the keywords generated by each model and the author keywords. (Click on papers' titles (first row) to access the full PubMed publications.)

| Increased cognition connectivity network in major depression disorder: a FMRI study | | | Characterization of Chromoshadow Domain-mediated Binding of Heterochromatin Protein 1 α (HP1 α) to Histone H3 | | |
|---|---|--|--|--|--|
| Author keywords | EmbedRank | docT5keywords | Author keywords | EmbedRank | docT5keywords |
| Depression, First episode, Cognition connectivity network, Functional magnetic resonance imaging, Resting state | cingulate gyrus, parietal cortex, right dorsolateral prefrontal cortex, precentral gyrus, frontal gyrus, abnormal cognition connectivity network, fmri, middle frontal gyrus, dlpc, cognitive dysfunction | psychological distress, cognitive impairment, depression | Chromatin Remodeling, Protein/Protein Interactions, Chromoshadow Domain, Chromatin Structure, Histones | non-histone chromatin protein, chromatin binding, heterochromatin protein, chromatin, histone, dimerization, transcriptional regulation, residues, binding region, binding | transcription genetic, animals, dna binding proteins, h3 histones, heterochromatin protein 1 subunit, genes regulator, chromatin, molecular sequence data, dna binding, chromatin research |

We can see that docT5keywords gets good results on collections where documents length is either short or medium. However, although our model also performs better than EmbedRank on collections where documents size is large, its performance is poor compared with datasets with smaller documents. We attribute this behaviour of docT5keywords to the input token limitation of T5 both in training and inference. That limits our approach to only using abstracts (and even not the complete abstract sometimes) for the task. Therefore, as documents' abstracts in short and medium papers represent a more significant part of the document, docT5keywords performs better. We could try to alleviate this problem either by doing the inference process in more than one step, taking segments of the abstract, or using alternative Transformer architectures [27]. We leave these alternatives for future work.

Table 2 shows two examples of the keywords assigned by each model and by the authors to two CDS-2016 papers. Note that only a maximum of 10 keywords are shown per each method.

5.2. Online study

After showing how each model performs in offline evaluation, we will discuss the results obtained on the user-based assessment, comparing the scores assigned to each keyword set type and the correlation between online and offline tasks. Finally, we will also examine the

Table 3

Mean score for each keyword type by assessor.

| Keywords type | Assessor 1 | Assessor 2 | Assessor 3 | Assessor 4 | Mean |
|--|------------|------------|------------|------------|------|
| Author | 2.46 | 1.98 | 2.53 | 2.30 | 2.32 |
| EmbedRank++ s2v ($\lambda = 0.55$) | 2.42 | 1.82 | 2.60 | 2.49 | 2.33 |
| docT5keywords | 2.00 | 1.69 | 1.87 | 1.72 | 1.82 |
| docT5keywords (<i>only creative</i>) | 1.62 | 1.48 | 1.21 | 1.17 | 1.37 |
| docT5keywords (<i>only not creative</i>) | 2.58 | 1.96 | 2.66 | 2.79 | 2.49 |

agreement between assessors in their evaluations.

Table 3 shows the mean score for each model and author keywords based on the assessors’ evaluation. As we may see in the results, EmbedRank performs much better than docT5keywords. Moreover, EmbedRank gets better results than the author keywords overall, which is quite impressive.

If we transfer the offline task results to the user evaluation task, it is supposed that EmbedRank should perform slightly better than our model (docT5keywords) as the difference in the previous evaluation was as significant. We attribute this variance in the model’s performance to three aspects of the CDS experiments:

- First, we did not follow an early stopping approach to train the model on MAG that we used to test on CDS. We merely used two epochs given the larger size of the training and the implications on training times. This, of course, may result in insufficient model training.
- Second, we used datasets of different nature to train and test the model. The shift between collections’ domains may result in worse results, especially when generating creative keywords.
- Third, offline evaluation follows a binary evaluation approach, while online evaluation follows a graded relevance approach. That means that offline evaluation penalizes equally both keywords that are relatively close to the document’s content (even present in it) but not in the ground truth and creative keywords that are far from representing the document. Meanwhile, the online evaluation methodology will penalize much more hallucinated creative keywords than those representing the document but missing in the golden truth.

Further elaborating on the last point, the concept of “hallucination” [28] stands for the problem that NLG systems generate texts that say false or not in accordance with the input data. Models that produce innovative results may generate keywords not present in the ground truth that either fit the record or not. In this case, as we can see by the relative scores of creative and not creative keywords generated by docT5keywords, our model tends to produce low-quality creative keywords. Hallucination is a well-known problem in other areas, such as neural approaches to image captioning [29]. However, recent work in NLP and NLG systems suggest

Table 4
Pairwise inter-assessor agreement I (A: Assessor).

| | A1 vs A2 | | | | A1 vs A3 | | | | A1 vs A4 | | | | | | |
|---------------------|----------|----|-----|-----|----------|---|----|----|----------|-----|---|----|----|----|-----|
| | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | | | |
| Raw scores | 0 | 22 | 40 | 24 | 8 | 0 | 42 | 11 | 17 | 24 | 0 | 28 | 22 | 17 | 27 |
| | 1 | 1 | 35 | 19 | 4 | 1 | 1 | 8 | 23 | 27 | 1 | 4 | 16 | 12 | 27 |
| | 2 | 9 | 48 | 30 | 21 | 2 | 10 | 9 | 24 | 65 | 2 | 4 | 22 | 16 | 66 |
| | 3 | 9 | 124 | 183 | 124 | 3 | 34 | 16 | 47 | 343 | 3 | 11 | 55 | 37 | 337 |
| Cohen’s Kappa Score | 0.085 | | | | 0.253 | | | | 0.211 | | | | | | |

Table 5
Pairwise inter-assessor agreement II (A: Assessor).

| | A2 vs A3 | | | | A2 vs A4 | | | | A3 vs A4 | | | | | | |
|---------------------|----------|----|----|----|----------|---|----|----|----------|-----|---|----|----|----|-----|
| | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | | | |
| Raw scores | 0 | 22 | 6 | 9 | 4 | 0 | 16 | 11 | 7 | 7 | 0 | 28 | 42 | 13 | 4 |
| | 1 | 32 | 28 | 61 | 126 | 1 | 16 | 56 | 43 | 132 | 1 | 6 | 17 | 10 | 11 |
| | 2 | 22 | 8 | 33 | 193 | 2 | 8 | 37 | 28 | 183 | 2 | 7 | 31 | 25 | 48 |
| | 3 | 11 | 2 | 8 | 136 | 3 | 7 | 11 | 4 | 135 | 3 | 6 | 25 | 34 | 394 |
| Cohen’s Kappa Score | 0.102 | | | | 0.113 | | | | 0.369 | | | | | | |

that it is also a relevant problem in many of these models applications. For example, it is not acceptable to label a document with an incorrect keyword in our particular case, as it may cause misleading results in later tasks. As shown in this recent survey [28], several ideas were proposed to reduce hallucination following diverse techniques and in different areas of NLG. The best way to address this problem is to do proper human evaluations, which should spot this model’s issue (like it did in our user study). Unfortunately, user-based assessments are costly. Thus, some proposals have been made to adapt classic evaluation datasets to address this problem [28].

To finish models’ performance comparison, after spotting the hallucination problem of our model, we decided to compute our model’s scores using only not creative keywords (filtering the generated keywords which cannot be found in the document). As shown in Table 3, our model under this scenario outperforms EmbedRank and even author keywords.

Closing this section, we present the results when evaluating the agreement between assessors. The first step into this task was the pairwise study of the inter-assessor agreement. We summarise the results of this study in Table 4 and Table 5. The "raw scores" section shows the 4x4 confusion matrices for each pair of assessors. These matrices represent how each assessor responded in contrast to the others. Note that the diagonal of the matrix represents the number of matches each pair of assessors had.

To finish this pairwise study, we computed Cohen’s Kappa score (κ), which allows us to measure the concordance grade between each pair of assessors. We can see the κ scores for the pairs between assessors 1, 3 and 4 are higher than with assessor 2, which means the latter does not

agree much with the others.

To wrap up the user study, we compute the unified concordance grade between all assessors by a naive mean of the Cohen's Kappa score per assessors' pair. We also used Fleiss' Kappa, an alternative to Cohen's Kappa that allows working with more than two assessors to compute the global concordance rate. Results obtained for these metrics were reasonably similar, having **0.193** for Cohen's Kappa and **0.177** for Fleiss' Kappa. However, these results were lower than expected because of the high disagreement between assessor 2 and the other assessors. If we ignore assessor 2, the values are as follows: **0.289** for average Cohen's Kappa and **0.289** for Fleiss' Kappa. Having this in mind, we can conclude that evaluations have agreement enough (maybe ignoring assessor 2) to say that the user study results can be adequately considered.

6. Conclusions and future work

This paper explored the potential of text-to-text transfer transformers (T5) in the keyword labelling for scientific documents task, comparing it to a state-of-the-art model like EmbedRank. Our model, docT5keywords, outperforms EmbedRank on classic datasets having an outstanding performance on collections formed by short and medium-size papers.

We also demonstrated the problems of classic datasets when evaluating models which rely on creativity by showing the performance decrease of our model in a user-based evaluation. This evaluation showed the hallucination problems that NLG models may have. Despite this, we have to highlight the good performance figures of our model to extract the best keywords from the text.

In our future work, we plan to delve into the hallucination issue that our model has and work on correctly assessing creative models in an offline task.

Acknowledgments

This work was supported by projects PLEC2021-007662 (MCIN/AEI/10.13039/501100011033, Ministerio de Ciencia e Innovación, Agencia Estatal de Investigación, Plan de Recuperación, Transformación y Resiliencia, Unión Europea-Next Generation EU) and RTI2018-093336-B-C22 (Ministerio de Ciencia e Innovación, Agencia Estatal de Investigación). The first and third authors also thank the financial support supplied by the Consellería de Cultura, Educación e Universidade (GPC ED431B 2022/33). The first author also acknowledges the support of grant DIN2020-011582 financed by the MCIN/AEI/10.13039/501100011033.

References

- [1] G. Marchionini, Exploratory search: from finding to understanding, *Communications of the ACM* 49 (2006) 41–46.
- [2] E. L.-C. Law, V. Roto, M. Hassenzahl, A. P. Vermeeren, J. Kort, Understanding, scoping and defining user experience: a survey approach, in: *Proceedings of the SIGCHI conference on human factors in computing systems*, 2009, pp. 719–728.

- [3] R. Guha, R. McCool, E. Miller, Semantic search, in: Proceedings of the 12th international conference on World Wide Web, 2003, pp. 700–709.
- [4] D. Tunkelang, Faceted search, Synthesis lectures on information concepts, retrieval, and services 1 (2009) 1–80.
- [5] E. N. Efthimiadis, Query expansion., Annual review of information science and technology (ARIST) 31 (1996) 121–87.
- [6] M. Steinbach, G. Karypis, V. Kumar, A comparison of document clustering techniques (2000).
- [7] Y. H. Li, A. K. Jain, Classification of text documents, The Computer Journal 41 (1998) 537–546.
- [8] Y. Li, M. Yang, Z. M. Zhang, Scientific articles recommendation, in: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, CIKM '13, Association for Computing Machinery, New York, NY, USA, 2013, p. 1147–1156. URL: <https://doi.org/10.1145/2505515.2505705>. doi:10.1145/2505515.2505705.
- [9] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (2009) 1345–1359.
- [10] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020) 1–67. URL: <http://jmlr.org/papers/v21/20-074.html>.
- [11] K. Bennani-Smires, C. Musat, A. Hossmann, M. Baeriswyl, M. Jaggi, Simple unsupervised keyphrase extraction using sentence embeddings, in: Proceedings of the 22nd Conference on Computational Natural Language Learning, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 221–229. URL: <https://aclanthology.org/K18-1022>. doi:10.18653/v1/K18-1022.
- [12] A. Hulth, Improved automatic keyword extraction given more linguistic knowledge, in: Proceedings of the 2003 conference on Empirical methods in natural language processing, 2003, pp. 216–223.
- [13] T. D. Nguyen, M.-Y. Kan, Keyphrase extraction in scientific publications, in: International conference on Asian digital libraries, Springer, 2007, pp. 317–326.
- [14] K. Wang, Z. Shen, C. Huang, C.-H. Wu, Y. Dong, A. Kanakia, Microsoft academic graph: When experts are not enough, Quantitative Science Studies 1 (2020) 396–413.
- [15] E. Maddalena, K. Roitero, G. Demartini, S. Mizzaro, Considering assessor agreement in ir evaluation, in: Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval, 2017, pp. 75–82.
- [16] E. Papagiannopoulou, G. Tsoumakas, A review of keyphrase extraction, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 10 (2020) e1339.
- [17] F. Rousseau, M. Vazirgiannis, Main core retention on graph-of-words for single-document keyword extraction, in: European Conference on Information Retrieval, Springer, 2015, pp. 382–393.
- [18] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: E. P. Xing, T. Jebara (Eds.), Proceedings of the 31st International Conference on Machine Learning, volume 32 of *Proceedings of Machine Learning Research*, PMLR, Beijing, China, 2014, pp. 1188–1196. URL: <https://proceedings.mlr.press/v32/le14.html>.
- [19] M. Pagliardini, P. Gupta, M. Jaggi, Unsupervised learning of sentence embeddings us-

- ing compositional n-gram features, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) (2018). URL: <http://dx.doi.org/10.18653/v1/N18-1049>. doi:10.18653/v1/n18-1049.
- [20] J. Carbonell, J. Goldstein, The use of mmr, diversity-based reranking for reordering documents and producing summaries, in: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, 1998, pp. 335–336.
- [21] X. J. Zhu, Semi-supervised learning literature survey (2005).
- [22] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: <https://aclanthology.org/N19-1423>. doi:10.18653/v1/N19-1423.
- [23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).
- [24] J. L. Ba, J. R. Kiros, G. E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450 (2016).
- [25] N. J.-M. Blackman, J. J. Koval, Interval estimation for cohen’s kappa as a measure of agreement, Statistics in medicine 19 (2000) 723–741.
- [26] J. L. Fleiss, B. Levin, M. C. Paik, et al., The measurement of interrater agreement, Statistical methods for rates and proportions 2 (1981) 22–23.
- [27] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The long-document transformer, arXiv:2004.05150 (2020).
- [28] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. Bang, A. Madotto, P. Fung, Survey of hallucination in natural language generation, arXiv preprint arXiv:2202.03629 (2022).
- [29] A. Rohrbach, L. A. Hendricks, K. Burns, T. Darrell, K. Saenko, Object hallucination in image captioning, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 4035–4045. URL: <https://aclanthology.org/D18-1437>. doi:10.18653/v1/D18-1437.