Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 6663–6671
Marseille, 11–16 May 2020

Two Huge Title and Keyword Generation Corpora of Research Articles

Erion Çano, Ondřej Bojar
Institute of Formal and Applied Linguistics, Charles University
Prague, Czech Republic
{cano, bojar}@ufal.mff.cuni.cz

Abstract

Recent developments in sequence-to-sequence learning with neural networks have considerably improved the quality of automatically generated text summaries and document keywords, stipulating the need for even bigger training corpora. Metadata of research articles are usually easy to find online and can be used to perform research on various tasks. In this paper, we introduce two huge datasets for text summarization (OAGSX) and keyword generation (OAGKX) research, containing 34 million and 23 million records, respectively. The data were retrieved from the Open Academic Graph which is a network of research profiles and publications. We carefully processed each record and also tried several extractive and abstractive methods of both tasks to create performance baselines for other researchers. We further illustrate the performance of those methods previewing their outputs. In the near future, we would like to apply topic modeling on the two sets to derive subsets of research articles from more specific disciplines.

Keywords: text summarization, keyword generation, corpus construction, research articles, huge corpora

1. Introduction

The ongoing tendency towards data-driven solutions for more and more tasks such as MT (Machine Translation), TS (Text Summarization), KG (Keyword Generation), and other tasks related to natural languages has created incentives for crawling the Web to produce large text corpora of various types. Furthermore, recent open data initiatives of governments[1] and other institutions that encourage the publication of more data on the Web have induced the same effect. From academia, there are initiatives such as ArnetMiner (Tang et al., 2008) that try to integrate existing scientific data from various resources in common networks for easier retrieval and exploitation. Among the various types of texts published in the Web, the metadata of research articles (e.g., titles, abstracts, keywords, etc.) are probably the easiest to find in large quantities, since they are usually not restricted. In fact, small corpora of research articles were used since the 90s to explore extractive KG (Witten et al., 1999; Turney, 1999) and TS (Mani and Bloedorn, 1997; Goldstein et al., 2000) techniques.

Research on these tasks has switched from the extractive paradigm to the recent abstractive one that is based on sequence-to-sequence learning with neural networks. The respective models are usually data-hungry, emphasizing the need for larger corpora in both TS and KG tasks. In this paper, we first review the popular existing datasets used for TS and KG research. We later describe the processing steps we followed, starting from the retrieval of ArnetMiner OAG (Open Academic Graph) data collection to the creation of two novel and huge corpora: OAGSX[2] and OAGKX.[3] The first one contains more than 34 million records consisting of paper abstracts and titles. It is suitable for TS experiments (more specifically for title generation which is a form of TS). The second one contains roughly 23 million abstracts, titles, and lists of keywords and is best suited for KG experiments. The data samples in the two corpora were carefully examined and various statistics about the text lengths and the lexical similarities between abstracts, titles, and keywords are presented.

We also explored the performance scores (ROUGE for TS and F_1@k for KG) of existing extractive and abstractive TS and KG solutions, trying them on evaluation minisets derived from OAGSX and OAGKX. According to our results, the recent abstractive methods based on sequence-to-sequence learning take a considerable time to train but perform better than the extractive methods on both tasks. To the best of our knowledge, our two data collections are the biggest of their kind that can be found online for free. We release them under the Creative Commons By 4.0 License. As future work, we would like to perform topic recognition on the articles of the two collections. This may lead to the creation of many subsets of research articles data from more specific scientific disciplines.

2. Background

2.1. Text Summarization

Automatic TS research explores intelligent methods to compress text documents into shorter summaries that express the main ideas of the source. It is mostly driven by our need to have shorter and easy-to-read summaries of long documents for saving reading time. Sometimes, we also need to have summaries of conversation threads (e.g., emails or chat messages). Multi-document TS is important when we want concise information from a set of documents and summaries of conclusions from meetings (minuting) or other event discussions. Another type of summarization aims to create short client reviews about different aspects of certain products or services. Title generation is yet another form of TS which is about paraphrasing the content of a text to produce an appropriate title for it.

There are two fundamental approaches for performing TS. The extractive way tries to select the most important and relevant parts from the source document and combines them to produce a shorter summary which is concise, co-

[1] https://www.data.gov/open-gov
[2] http://hdl.handle.net/11234/1-3079
[3] http://hdl.handle.net/11234/1-3062

herent and readable. In this case, the target or output text contains verbatim copies of words or phrases taken from the source or input. The abstractive approach, on the other hand, learns to paraphrase the information required for the summary, instead of directly copying it from the source. This is somehow better, but the methodology is more complex and requires more resources. The TS research of the 90s and early 00s was mostly based on extractive methods. The respective techniques used unsupervised learning (Goldstein et al., 2000; Barzilay and Elhadad, 1997), supervised learning (Wong et al., 2008; Fukumoto, 2004) or graph methods (Erkan and Radev, 2004; Mani and Bloedorn, 1997) to select the most important lexical units from the source documents.

The abstractive approach has become popular in recent years, following the progress in sequence-to-sequence learning with neural networks (the encoder-decoder framework). LSTM neural networks (Hochreiter and Schmidhuber, 1997) are combined and enhanced with advanced mechanisms like the attention of Bahdanau et al. (2015) for a more effective learning of the alignments between the text sequences. Attention allows the model to focus on different segments of the input during generation and was successfully used by Rush et al. (2015) to summarize news articles. The problem of unknown words (not seen in source texts) was also mitigated by the copying technique (Gu et al., 2016; Gulcehre et al., 2016). Furthermore, the coverage (Tu et al., 2016a) and intra-attention (Paulus et al., 2017) mechanisms were proposed to alleviate word repetitions in the summaries, a notorious problem of the encoder-decoder models.

Scoring results were pushed even further just recently by mixing reinforcement learning concepts such as policy gradient (Rennie et al., 2017) into the encoder-decoder architecture. It optimizes the learning objective (higher summarization score) and still keeps an appropriate quality of the produced summaries. A recent performance comparison of various abstractive TS methods can be found in Çano and Bojar (2019a).

2.2. Keyphrase Generation

Keyphrase generation is the process of analyzing a document and producing sets of one or a few words (keywords or keyphrases, used interchangeably) that best represent its main concepts or topics. These keywords are frequently utilized nowadays to annotate digital objects (e.g., research articles, books, product descriptions, etc.) and quickly find them in digital libraries, online stores, etc. A keyword string is a concatenation of several keywords (commas or semicolons are typically used as separators) attached to one of those objects. The need to process large amounts of documents with missing keywords created incentives for research in automatic KG since the 90s. The popular supervised learning algorithms of that time were used by authors like Turney (2000) or Witten et al. (1999) in combination with lexical features to extract keywords from the documents. Furthermore, graph-based methods (Rose et al., 2010; Wan and Xiao, 2008) or other unsupervised KG methods (Campos et al., 2018; Nart and Tasso, 2014) were proposed later in the 00s.

The above extractive KG solutions were very successful because of their simplicity and execution speed. However, extractive KG suffers from a serious inherent handicap: its inability to produce absent keywords (keywords not appearing in the source text). Meng et al. (2017) analyzed the author's keywords in popular corpora. They observed that absent and present (keywords that also appear in the source text) keywords assigned by paper authors are almost equally frequent. It is thus a serious drawback to completely ignore the absent keywords.

The recent advances in language representation (Mikolov et al., 2013; Pennington et al., 2014) and sequence-to-sequence learning (Bahdanau et al., 2015; Vaswani et al., 2017) motivated several researchers like Meng et al. (2017) or Zhang and Xiao (2018) to explore abstractive KG in the context of the encoder-decoder framework. The encoder-decoder network structures were initially utilized to perform MT and got quick adoption on similar tasks like TS and KG that are also based on the sequence-to-sequence transformation between source and target texts. Furthermore, same as in TS research, various reinforcement learning concepts like adaptive rewards that are being explored are raising the performance scores even higher (Chan et al., 2019). Abstractive KG is now a vibrant research direction with more than a dozen of publications only in the last three years. More comprehensive surveys of KG literature can found in other recent publications like (Papagiannopoulou and Tsoumakas, 2019) and (Çano and Bojar, 2019b).

2.3. Scientific Article Data Sources

The current hype of deep neural networks has created strong incentives for producing data collections by crawling the web. The richest sets of language resources are used for machine translation (Resnik and Smith, 2003; Tiedemann, 2012; Mahata et al., 2016; Shi et al., 2005) and for sentiment analysis (Bosco et al., 2013; Çano and Bojar, 2019c; Maas et al., 2011; Çano and Morisio, 2019; Jiménez Zafra et al., 2015). They are mostly driven by the information technology giants that continuously improve their language-related applications and marketing companies to understand customers' perceptions about various online products.

TS and KG research of the 90s and early 00s was mostly based on extractive methods that did not rely on big training corpora. Things gradually changed in the late 00s with the rising popularity of the encoder-decoder framework. The current TS and KG methods are also highly dependent on the big language corpora since they are mainly based on sequence-to-sequence learning with neural networks. Some of the most popular corpora in TS and KG literature are presented in Table 1. One of the first big datasets was the annotated English Gigaword (Napoles et al., 2012) used for abstractive TS by Rush et al. (2015). It contains about nine million news articles and headline summaries. Each headline was paired with the first sentence of the corresponding article to create the training base for the experiments.

Newsroom (Grusky et al., 2018) is a very recent and heterogeneous bundle of about 1.3 million news articles. It contains writings published from 1998 to 2017 by 38 major newsrooms. Another recent dataset of news articles

Reference	Name	Content	# Docs
Napoles et al. (2012)	Gigaword	News	9 M
Grusky et al. (2018)	Newsroom	News	1.3 M
Nallapati et al. (2016)	CNN/DM	News	287 K
Hyperlink	DUC-2004	News	500
Hulth (2003)	Inspec	Papers	2000
Krapivin et al. (2010)	Krapivin	Papers	2304
Kim et al. (2010)	SemEval	Papers	244
Meng et al. (2017)	KP20k	Papers	567 K
Nikolov et al. (2018)	tit-gen	Papers	900 K
Nikolov et al. (2018)	abs-gen	Papers	5 M

Table 1: Summary and keyword generation datasets

is CNN/Dailymail of Nallapati et al. (2016). It has become the most popular corpus for text summarization experiments. This dataset provides a rich collection of news articles and the corresponding multi-sentence summaries (news highlights). It is thus very suitable for training and testing summarization models of longer texts. DUC-2004 is another dataset that was originally created for the Document Understanding Conference.[4] It has been mostly used as an evaluation baseline, given its small size. It consists of 500 document-summary pairs curated by human experts.

Besides using news articles, it is also possible to exploit texts of scientific articles for TS research. In fact, those kinds of texts have been used since long ago to conduct KG research. There are many relatively small datasets of scientific publications and the corresponding keywords that have been used for many years to test extractive or graph-based KG methods. One of the most popular KG datasets is Inspec released by Hulth (2003). It consists of 2000 paper titles (1500 for training and 500 for testing), abstracts and keywords from journals of Information Technology, published from 1998 to 2002.

Krapivin et al. (2010) released another collection of papers that has been frequently used in the literature. It consists of 2304 computer science papers published by ACM from 2003 to 2005. The advantage of this dataset is the availability of the full paper texts together with the corresponding metadata. A smaller dataset is SemEval of Kim et al. (2010) that was originally created for the Semantic Evaluation task. It contains 244 papers that belong to conference and workshop proceedings.

A few years ago, Meng et al. (2017) released KP20k which is today the most popular KG dataset. It contains 567830 Computer Science articles, 527830 used for training, 20 K for validation and 20 K for testing. This dataset has been used for training and comparing various recent abstractive KG methods. Nikolov et al. (2018) raised the data sizes even more by retrieving many scientific papers from libraries of biomedical research.[5] The authors derived and released two big (900 K and 5 M) corpora for TS (predicting abstracts from paper bodies) and title generation (predicting titles from abstracts).

Crawling public digital libraries or websites for text resources is an ongoing trend. ArnetMiner (Tang et al., 2008) is an initiative to integrate scientific data (publications, researcher profiles and more) from various resources in a common and unified network. A derivative product is the OAG data collection of scientific publications (Sinha et al., 2015). Each record is a JSON line with publication metadata like *authors*, *title*, *abstract*, *keywords*, *year* and more. In the following section, we describe the processing steps we performed on OAG collection to derive OAGSX and OAGKX datasets.

Attribute	Title	Abstract
Total	449 M	6 B
Min / Max	3 / 25	50 / 400
Mean (Std)	13.1 (5.1)	182.2 (89.2)
Jindex	6.7 % (3.9 %)	
Overlap	77 % (18 %)	
Total size	34 408 509 title-abstracts	

Table 2: Token statistics of OAGSX

Attribute	Title	Abstract	Keywords
Total	290 M	4 B	270 M
Min / Max	3 / 25	50 / 400	2 / 60
Mean (Std)	12.8 (4.9)	175.1 (86.5)	11.9 (7.5)
Jindex	7.1 % (4 %)	6 % (4.8 %)	
Overlap	78 % (17 %)	68 % (25 %)	
Total size	22 674 436 title-abstract-keywords		

Table 3: Token statistics of OAGKX

3. OAGSX and OAGKX Corpora

For producing large TS and KG text collections, we utilized the text fields of the OAG bundle. From that same article set, we filtered the records containing at least the *title* and the *abstract* for OAGSX and those with the *title*, *abstract*, and *keywords* for OAGKX. We dropped the duplicate entries in each of our two collections. As a result, the samples inside each of the corpora are unique (there is still overlapping between OAGSX and OAGKX samples, since they were both derived from the OAG collection). An automatic language identifier[6] was used to remove the records with abstracts not in English. We also cleared the messy symbols and lowercased everything. Finally, Stanford CoreNLP (Manning et al., 2014) was used to tokenize the *title* and *abstract* texts.

After the preprocessing steps, we observed the size and token lengths of the records. Since there were many outliers (e.g., records with very long or very short abstracts), we removed all records with a title not in the range of 3-25 tokens and abstract not within 50-400 tokens. In the case of OAGKX, we also removed samples with keyword string not in the range of 2-60 tokens or 2-12 keywords. After this, OAGSX was reduced to a total of about 34.4 million records. OAGKX, on the other hand, shrank to about 22.6 million records.

[4] https://duc.nist.gov/duc2004/

[5] https://www.nlm.nih.gov

[6] https://pypi.org/project/langid

Attribute	Value
Total	133 295 056
Min / Max	2 / 12
Mean (Std)	5.9 (3.1)
Present	52.7 % (28.3 %)
Absent	47.3 % (28.3 %)

Table 4: Keyword statistics of OAGKX

Some further statistics of the two final datasets are presented in Tables 2 and 3. In the case of OAGSX, the average title and abstract lengths are about 13.1 and 182.2 tokens respectively (standard deviation is always given in parenthesis). The corresponding values in OAGKX are 12.8 and 175.1 (slightly lower). For OAGKX we also see that the keyphrase strings contain 11.9 tokens on average. We also wanted to observe the lexical similarity between the titles and abstracts. One way for this is to compute the Jaccard similarity (Jindex in Tables 2 and 3) of the whole token sets using the following equation:

$$J(A, B) = \frac{|T \cap A|}{|T \cup A|} = \frac{|T \cap A|}{|T| + |A| - |T \cap A|} \quad (1)$$

where T is the set of unique tokens in the title and A is the set of unique tokens in the abstract. In OAGSX, the Jaccard similarity between abstracts and titles is 6.7 %. In OAGKX, it is 7.1 % between the abstracts and titles and 6 % between abstracts and keyword strings. Another indicator is the overlap $o(s, t) = \frac{|\{s\} \cap \{t\}|}{|\{t\}|}$ which represents the fraction of unique target tokens t (e.g., in the *title* or in the *keyword string* excluding punctuation symbols) that overlap with a source token (e.g., in *abstract*) s. The overlaps between titles and abstracts are very similar (77 % and 78 %) in both datasets. In the case of OAGKX, the overlap between abstracts and keyword strings is 68 %.

We further analyzed the keyword distribution in OAGKX (Table 4). There is a total of about 133 million keywords, with an average of 5.9 keywords per article. In abstractive KG experiments, it is also important to know the distribution of present and absent keywords. The present rate $p(s, k) = \frac{|k \cap s|}{|k|}$ is the fraction of the keywords k that also appear in the source text s. This is similar to the overlap, with the difference that there might be token repetitions within each counted keyword. The absent rate $a(s, k) = \frac{|k| - |k \cap s|}{|k|}$ is its complement or the fraction of keywords k that do not appear in the source text s. From Table 4 we see that the present and absent keywords in OAGKX are almost evenly distributed (52.7 % and 47.3 % each). This observation is in line with that of Meng et al. (2017), emphasizing once again the importance of the absent keywords.

Another interesting exploration we wanted to perform was the identification of the topics (or research domains) in each dataset record to report the corresponding statistics. This could lead to the creation of many subsets of OAGSX and OAGKX with scientific articles from more specific disciplines (clustering together the articles from the same research direction). Unfortunately, topic modeling was not easy to perform on OAGSX and OAGKX, given the huge size of the two corpora and our limited computational resources. It thus remains a potential future work.

We still inspected a few of the samples from each dataset manually. Their texts mostly belong to papers from biomedical disciplines but there are also papers about psychology, geology, or various technical directions. To our best knowledge, OAGSX and OAGKX are the largest available collections of scientific paper metadata that can be used for TS and KG experiments. Their importance is thus twofold: (i) They can supplement existing collections if more training samples are required. (ii) They can serve as sources for deriving article subsets of more specific scientific disciplines or domains.

4. Evaluation Experiments

We tried various extractive and abstractive methods for TS and KG on evaluation subsets from two corpora. In the following sections, we report the achieved performance scores of the automatic evaluation process. We also illustrate the output of each method with examples.

4.1. Title Generation

For the title generation experiments, we formed three evaluation subsets from OAGSX: a training set of 1 million samples, a validation set of 10 thousand samples and a test set of 10 thousand samples. To reduce the vocabulary size (important for abstractive text summarizers), we further replaced number patterns with the # symbol in each of them. The most simple and raw baseline we used is Random-k (Random-1 in our case) which splits the source text into sentences and randomly picks k of them as its summary. In our case, since we are generating the title of the articles, we randomly pick only one of the abstract sentences as the predicted title. Random-1 can be considered as the lowest scoring boundary since it uses no intelligence at all. Another popular baseline is Lead-k (Lead-1 in our case). It is based on the concept of “summary lead”, which concisely explains the main idea of a text in its first sentence or first few sentences. Lead-1 picks the first sentence from the source text to generate its title.

LexRank is a stochastic graph-based method for assessing the importance of textual units in a source text (Erkan and Radev, 2004). When used to perform extractive TS, it computes the importance of those units using the concept of eigenvector centrality in the graph. The top k units (the top sentence in this case) are returned as the best summary of the document.

One of the abstractive text summarizers we used is PointCov of See et al. (2017) which is based on the encoder-decoder framework. In each decoding step, it implements the pointing/copying mechanism (Gu et al., 2016; Gulcehre et al., 2016) to compute a generation probability. The latter is used to decide whether the next word should be predicted or directly copied from the source sequence. Another feature is the implementation of the coverage mechanism (Tu et al., 2016a) which helps to avoid word repetitions in the target sequence. We trained PointCov with a hidden layer of 256 dimensions and word embeddings of 128 dimensions.

The other abstractive summarizer we picked is the Transformer model that represents one of the most important achievements in sequence-to-sequence learning of the last years (Vaswani et al., 2017). It is totally based on the attention mechanism, removing all recurrent or convolutional structures. Although it was primarily designed for MT, the Transformer can also work for text summarization. It basically learns the alignments between the input (source) texts and the output (target) summaries. As documented by Çano and Bojar (2019a), Transformer reveals the highest data efficiency scores on the popular TS datasets. We used the Transformer model with four layers in both encoder and decoder blocks, 512 dimensions in each layer, including the embedding layers, 200 K training steps, and 8000 warm-up steps. Both PointCov and Transformer were trained with Adam optimizer (Kingma and Ba, 2014) using $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ and mini-batches of 16 training samples. We used two NVIDIA GTX 1080Ti GPUs at once for the training process.

Random-1, Lead-1, and LexRank (the three extractive methods) were directly applied in the test set of 10 thousand examples. For PointCov and Transformer, we used all the three evaluation subsets. ROUGE-1, ROUGE-2, and ROUGE-L scores (Lin, 2004) were computed by comparing the title outputs of each method with the titles of the original papers. The results are presented in Table 5.

As we expected, Random-1 is the worst in all three ROUGE scores. Lead-1 performs well, reaching a peak score of 33.8 % in ROUGE-1. It is actually slightly better than LexRank in all the three metrics. Transformer and PointCov, which are the two abstractive neural networks we tried, perform better than the three extractive methods. They achieve similar results, but the Transformer leads with a peak score of 37.27 % in ROUGE-1. It is also important to note that the three extractive methods took only a few minutes to produce the outputs. PointCov and Transformer, on the other hand, required 3 – 4 days for the training.

An abstract, its author's title, and the titles predicted by the above five methods are illustrated in Table 6. As we can see, all the methods have generated titles that are longer than those of the authors. The titles of Lead-1 and LexRank are very similar, both based on the first sentence of the abstract. The transformer has produced a very long title with an unfinished sentence. This problem could be fixed by using a lower value for the length of the target text. PointCov has generated a shorter sentence than the Transformer, but it is not very coherent.

Method	R_1	R_2	R_L
Random-1	22.67	8.02	18.44
Lead-1	33.83	16.8	28.14
LexRank	29.4	12.83	24.03
PointCov	36.12	18.88	30.21
Transformer	37.27	19.12	30.78

Table 5: Results on OAGSX

Abstract: the central bank 's lender of last resort role was developed by a series of authors in the very late eighteenth and through the nineteenth centuries . it was tested in practice in a number of countries and was found to be effective in providing monetary stability in the face of adverse shocks . there have recently been attempts to broaden the role to make the central bank responsible for the stability of asset markets , or for protecting individual banks and there have recently also been claims that an international lender of last resort is necessary . this article considers and rejects these proposed extensions to the classic lender of last resort role
Author's title: the lender of last resort reconsidered
Random-1 title: this article considers and rejects these proposed extensions to the classic lender of last resort role
Lead-1 title: the central bank 's lender of last resort role was developed by a series of authors in the very late eighteenth and through the nineteenth centuries
LexRank title: the central bank 's lender of last resort role was developed in the late eighteenth and through the nineteenth centuries
PointCov title: the central bank 's lender and its implications for the stability of asset comparative analysis
Transformer title: the central bank 's lender of last resort role and its implications for the stability of asset markets : a comparative analysis of the central and in the lender of the

Table 6: KE scores on OAGKX

Method	$\mathbf{F_1@5}$	$\mathbf{F_1@7}$	$\mathbf{F_1@10}$
TopicRank	17.12	20.81	20.75
RAKE	16.36	18.84	18.91
Maui	24.58	23.49	23.6
CopyRNN	28.15	28.93	28.96
CovRNN	27.76	29.15	29.04

Table 7: KE scores on OAGKX

4.2. Keyphrase Generation

We ran similar experiments on three evaluation sets derived from OAGKX: a training set of 631705 samples, a validation set of 10 thousand samples and a test set of 10 thousand samples. Once again, we tried and compared both extractive and abstractive KG methods. We used TopicRank of Bougouin et al. (2013) which is a popular graph-based extractive method that makes use of the PageRank algorithm (Brin and Page, 1998). It first uses clustering to group lexical units of the same topic. Then, it uses the graph-based ranking algorithm to score each topic cluster that is formed. At the end, one keyword is picked from each of the ranked clusters.

RAKE proposed by Rose et al. (2010) is one of the fastest available methods for extractive KG. It first removes punctuation symbols together with the stop words of the specified language and then creates a graph of word co-occurrences. Candidate words or phrases are scored based

Abstract: a complex polysaccharide accumulation was observed in the central nervous system (cns) of rats treated with d-penicillamine similar to lafora-like bodies . they have histochemical similarities comparable to bodies described in previous studies of lafora disease . the clinical usefulness of d-penicillamine has been limited by many side effects including renal damage . it is suggested that , in addition to d-penicillamine nephropathy , there are toxic effects of this drug on the cns
Title: polysaccharide accumulation in the central nervous system of d-penicillamine treated rats
Author's keywords: polysaccharide , central nervous system , side effect , d-penicillamine , lafora-like bodies , nephropathy
TopicRank keywords: central nervous system , d-penicillamine , accumulation , polysaccharide accumulation , cns , d-penicillamine effect , drug , lafora bodies , clinical , rats
RAKE keywords: clinical , polysaccharide , central nervous , rats , polysaccharide accumulation , lafora disease , renal damage , accumulation , lafora, cns
Maui keywords: central , central system , system , d-penicillamine , polysaccharide accumulation , polysaccharide , accumulation , lafora-like , lafora, bodies
CopyRNN keywords: central nervous , d-penicillamine , side effect , side , newborn rats , rat ileostomy , pregnant rats, nephropathy , mortality of rats , mortality
CovRNN keywords: side effect , central nervous , polysaccharide , rats , lafora bodies , polysaccharide effect , d-penicillamine , polysaccharide-derived , albino rats , trinitrobenzene sulfonic acid

Table 8: KE scores on OAGKX

on the degree and frequency of each word vertex in the graph. The k top-scoring candidates are returned as keywords. We also used Maui (Medelyan, 2009), a supervised extractive method that uses lexical features and bagged decision trees to predict whether a candidate phrase is a keyword or not.

CopyRNN (Meng et al., 2017) was the first abstractive KG method based on the encoder-decoder framework. Authors implemented the copying mechanism to balance between extracting present phrases from the source text with the generation of absent phrases. This work was followed by several recent studies that improve KG with various additional mechanisms. Finally, the last method we tried is CovRNN (Zhang and Xiao, 2018) which is very similar to CopyRNN. It tries to avoid the repeated keywords during generation by considering the correlation between the produced keywords at each generation step. This is achieved by implementing the coverage mechanism of Tu et al. (2016b).

We applied TopicRank and RAKE on the test set of 10 thousand records. Because of its memory limitations, Maui was trained on the first 30 thousand samples from the training set and tested on the test set. For CopyRNN and CovRNN, we used the full sizes of the three evaluation sets. For the comparison, we used F_1 scores of the full matches between the author's keywords and the top k keywords returned by each method. Given that each data sample has a variable-length keyword string, we picked the values 5, 7 and 10 for the k parameter. The obtained results are shown in Table 7. The first thing we can notice from the results is the fact that F_1@7 and F_1@10 scores are very similar to each other in each case. This is probably because few data samples contain more than 7 keywords in their keyword string (the average was 5.9). We also see that CopyRNN and CovRNN perform significantly better than the first three extractive methods. They achieve very similar scores in the three metrics. The peak score of 29.15 % is reached by CovRNN on F_1@7. From the three extractive methods, Maui performs better than the other two. TopicRank performs slightly better than RAKE. Once again, the training of the abstractive methods based on neural networks took about 3 days whereas the results of the extractive approaches (with the exception of Maui which was trained in few hours) were obtained in few minutes.

The source texts and the produced keywords (top ten) of a data sample are shown in Table 8. Apparently, both extractive and abstractive predictions are grammatically correct. However, few of the generations represent full keyword matches. There is also a considerable number of partial matches. The first four methods have produced certain word repetitions. We can also observe some "novel" (thou incorrect) phrases like "mortality of rats" or "trinitrobenzene sulfonic acid" that are produced by CopyRNN and CovRNN.

5. Conclusion

Today, we can find uncountable research article data that are freely available in digital libraries. Many relatively small collections of those data are frequently used to run text summarization and keyword generation experiments. In this paper, we described the steps we followed to process Open Academic Graph data and prepare two huge corpora: OAGSX of more than 34 million abstracts and titles that can be used for text summarization and OAGKX of about 23 million abstracts, titles, and keyword strings that can be used for keyword generation. To our best knowledge, these corpora of scientific paper metadata are the biggest freely available online. We also performed several experiments applying extractive and abstractive TS and KG methods on their subsets to help establish performance benchmarks that could be valuable to other researchers. In the future, we plan to apply topic modeling on the two collections for deriving many subsets of research articles from more specific scientific disciplines.

6. Acknowledgements

This research work was supported by the project No. CZ.02.2.69/0.0/0.0/16_027/0008495 (International Mobility of Researchers at Charles University) of the Operational Programme Research, Development and Education, the project no. 19-26934X (NEUREM3) of the Czech Science Foundation and ELITR (H2020-ICT-2018-2-825460) of the EU.

7. Bibliographical References

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Barzilay, R. and Elhadad, M. (1997). Using lexical chains for text summarization. In *In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17.

Bosco, C., Patti, V., and Bolioli, A. (2013). Developing corpora for sentiment analysis: The case of irony and senti-tut. *IEEE Intelligent Systems*, 28(2):55–63, March.

Bougouin, A., Boudin, F., and Daille, B. (2013). Topicrank: Graph-based topic ranking for keyphrase extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 543–551. Asian Federation of Natural Language Processing.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, April.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A. M., Nunes, C., and Jatowt, A. (2018). Yake! collection-independent automatic keyword extractor. In Gabriella Pasi, et al., editors, *Advances in Information Retrieval*, pages 806–810. Springer International Publishing.

Çano, E. and Bojar, O. (2019a). Efficiency metrics for data-driven models: A text summarization case study. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 229–239, Tokyo, Japan, October–November. Association for Computational Linguistics.

Çano, E. and Bojar, O. (2019b). Keyphrase generation: A multi-aspect survey. In *Proceedings of the 25th Conference of Open Innovations Association FRUCT*, pages 85–94, Helsinki, Finland, Nov. 2019.

Çano, E. and Bojar, O. (2019c). Sentiment analysis of czech texts: An algorithmic survey. In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence - Volume 2: NLPinAI*, pages 973–979, Prague, Czech Republic. INSTICC, SciTePress.

Çano, E. and Morisio, M. (2019). A data-driven neural network architecture for sentiment analysis. *Data Technologies and Applications*, 53(1):2–19.

Chan, H. P., Chen, W., Wang, L., and King, I. (2019). Neural keyphrase generation via reinforcement learning with adaptive rewards. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2163–2174, Florence, Italy, July. Association for Computational Linguistics.

Erkan, G. and Radev, D. R. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Fukumoto, J. (2004). Multi-document summarization using document set type classification. In *Proceedings of the Fourth NTCIR Workshop on Research in Information Access Technologies Information Retrieval, Question Answering and Summarization, NTCIR-4, National Center of Sciences, Tokyo, Japan, June 2-4, 2004*.

Goldstein, J., Mittal, V., Carbonell, J., and Kantrowitz, M. (2000). Multi-document summarization by sentence extraction. In *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic Summarization - Volume 4*, NAACL-ANLP-AutoSum '00, pages 40–48, Stroudsburg, PA, USA. Association for Computational Linguistics.

Grusky, M., Naaman, M., and Artzi, Y. (2018). Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 708–719. Association for Computational Linguistics.

Gu, J., Lu, Z., Li, H., and Li, V. O. (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)*, pages 1631–1640, Berlin, Germany, August. Association for Computational Linguistics.

Gulcehre, C., Ahn, S., Nallapati, R., Zhou, B., and Bengio, Y. (2016). Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149, Berlin, Germany, August. Association for Computational Linguistics.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780, Nov.

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223. Association for Computational Linguistics.

Jiménez Zafra, S. M., Berardi, G., Esuli, A., Marcheggiani, D., Martín-Valdivia, M. T., and Moreo Fernández, A. (2015). A multi-lingual annotated dataset for aspect-oriented opinion mining. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, Lisbon, Portugal.

Kim, S. N., Medelyan, O., Kan, M.-Y., and Baldwin, T. (2010). Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26. Association for Computational Linguistics.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. cite arxiv:1412.6980, Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Krapivin, M., Autayeu, A., Marchese, M., Blanzieri, E., and Segata, N. (2010). Keyphrases extraction from scientific documents. In Gobinda Chowdhury, et al., editors, *The Role of Digital Libraries in a Time of Global Change*.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*, page 10.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng,

A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June. Association for Computational Linguistics.

Mahata, S., Das, D., and Pal, S. (2016). WMT2016: A hybrid approach to bilingual document alignment. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 724–727, Berlin, Germany, August. Association for Computational Linguistics.

Mani, I. and Bloedorn, E. (1997). Multi-document summarization by graph search and matching. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, AAAI'97/IAAI'97, pages 622–628. AAAI Press.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Medelyan, O. (2009). *Human-competitive automatic topic indexing*. The University of Waikato, Phd Thesis.

Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., and Chi, Y. (2017). Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 582–592. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv e-prints*, Jan.

Nallapati, R., Zhou, B., dos Santos, C., Gulcehre, C., and Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290. Association for Computational Linguistics.

Napoles, C., Gormley, M., and Van Durme, B. (2012). Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, AKBC-WEKEX '12, pages 95–100, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nart, D. D. and Tasso, C. (2014). A domain independent double layered approach to keyphrase generation. In *WEBIST*.

Nikolov, N. I., Pfeiffer, M., and Hahnloser, R. H. R. (2018). Data-driven summarization of scientific articles. *CoRR*, abs/1804.08875.

Papagiannopoulou, E. and Tsoumakas, G. (2019). A review of keyphrase extraction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, n/a(n/a):e1339.

Paulus, R., Xiong, C., and Socher, R. (2017). A deep reinforced model for abstractive summarization. *CoRR*, abs/1705.04304.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. (2017). Self-critical sequence training for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195.

Resnik, P. and Smith, N. A. (2003). The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380.

Rose, S., Engel, D., Cramer, N., and Cowley, W. (2010). Automatic keyword extraction from individual documents. In Michael W. Berry et al., editors, *Text Mining. Applications and Theory*, pages 1–20. John Wiley and Sons, Ltd.

Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389. Association for Computational Linguistics.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1073–1083. Association for Computational Linguistics.

Shi, L., Niu, C., Zhou, M., and Gao, J. (2005). A dom tree alignment model for mining parallel data from the web. In *COLING-ACL*, January.

Sinha, A., Shen, Z., Song, Y., Ma, H., Eide, D., Hsu, B.-J. P., and Wang, K. (2015). An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 243–246, New York, NY, USA. ACM.

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., and Su, Z. (2008). Arnetminer: Extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 990–998, New York, NY, USA. ACM.

Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. In Nicoletta Calzolari (Conference Chair), et al., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Tu, Z., Lu, Z., Liu, Y., Liu, X., and Li, H. (2016a). Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 76–85. Association for Computational Linguistics.

Tu, Z., Lu, Z., Liu, Y., Liu, X., and Li, H. (2016b). Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany, August. Association for Computational Linguistics.

Turney, P. (1999). Learning to extract keyphrases from text.

Turney, P. D. (2000). Learning algorithms for keyphrase extraction. *Inf. Retr.*, 2(4):303–336, May.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In I. Guyon, et al., editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Wan, X. and Xiao, J. (2008). Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 855–860. AAAI Press.

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., and Nevill-Manning, C. G. (1999). Kea: Practical automatic keyphrase extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, DL '99, pages 254–255, NY, USA. ACM.

Wong, K.-F., Wu, M., and Li, W. (2008). Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 985–992, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhang, Y. and Xiao, W. (2018). Keyphrase generation based on deep seq2seq model. *IEEE Access*, 6:46047–46057.