Benchmarking Natural Language Processing Algorithms for Patent Summarization

Silvia Casola[1,*], Alberto Lavelli[1]

[1]University of Padua, Fondazione Bruno Kessler

Abstract

The number of patent applications is enormous, and patent documents are long and complex. Methods for automatically obtaining the most salient information in a short text would thus be useful for patent professionals and other practitioners. However, patent summarization is currently under-researched; moreover, the proposed methods are difficult to compare directly as they are generally tested on different datasets. In this paper, we benchmark several extractive, abstractive, and hybrid summarization methods on the BigPatent dataset, compare automatic metrics and show qualitative insights.

Keywords

Summarization, Patents, Natural language processing,, Natural language generation

1. Introduction

Patents protect inventions that their holders consider important enough to take legal action to obtain the monopoly in using, making, and selling them — and thus, profit from their wit. Thus, they help in valuing intellectual work. At the same time, inventors must disclose the invention and its characteristics in detail to file a patent application: thus, patents are intended to benefit society and help new knowledge spread — correcting the tendency to keep valuable technical details secret. Patents, however, are difficult to process: the number of patent applications is enormous and patent documents are long and hard to read, rich in technical and legal language.

To this end, tools that automatically extract or generate summaries from patent documents can be particularly valuable in helping patent agents, R&D groups, and other professionals; using summaries instead of the whole document can also improve the performance of automatic processes, as shown in other domains [1, 2].

In the general domain, summarization tools and methodologies have shown promising results; applications to the patent domain are, however, still relatively limited. Moreover, while previous work has explored methods for automatically generating patent summaries, these methods are hard to compare, as no generally accepted benchmarks exist; thus, conclusions on the pros and cons of each approach are hard to make. Even the most recent abstractive dataset presents important limitations and issues that might make direct comparisons meaningless [3].

To partially fill this gap, we benchmark existing approaches in the patent domain, specifically on the BigPatent [4] dataset. The dataset is popular in the NLP community, as patents present several challenges in terms of abstractivity, length, and language, among others; moreover, while not exempt from design issues, it is also one of the few patent benchmarks that allow for a direct comparison between approaches. We evaluate extractive, abstractive, and hybrid methods; we also explore transferring summarization methods from the scientific paper domain [5] with limited success. For each method, we discuss strengths and limitations, provide standard summarization metrics and qualitative insights.

PatentSemTech'23: 4th Workshop on Patent Text Mining and Semantic Technologies, July 27th, 2023, Taipei, Taiwan

*Corresponding author.

scasola@fbk.eu (S. Casola); lavelli@fbk.eu (A. Lavelli)

2. Previous work

2.1. Automatic text summarization

Methods for text summarization are generally classified into extractive, abstractive, and hybrid ones.

In extractive text summarization, a subset of sentences from the source document is chosen as the most representative, and the final summary is a simple concatenation of such sentences. Methods can be graph-based [6, 7, 8], rely on token frequency [9], or on learned intrinsic features [10, 11, 12].

In contrast, abstractive text summarization aims at generating a new piece of text based on the source, similar to what a person would do, and can contain novel vocabulary or expressions. Sequence-to-sequence models [13, 14, 15, 16, 17] are popular for this task, with transformer-based ones being particularly performative [18, 19, 20, 21]. Finally, hybrid methods try to fuse both approaches, for example, by extracting and rewriting sentences [22].

Extractive models are generally simpler than abstractive ones and require fewer computational resources and data; however, summaries have to contain complete sen-

tences from the source, which often contain both central and peripheral information. Moreover, the final summary is a simple concatenation of sentences, with pending references and no discourse structure. Abstractive summaries are more similar to those written by humans. Information can be easily condensed and the generated text is much more natural and easier to read. However, abstractive models might produce non-factual information, i.e. include statements that are not in the source or that directly contradict them. See, e.g., [23] for a comprehensive survey of summarization techniques.

2.2. Patent summarization

Many traditional approaches for patent summarization have been extractive. The document is often segmented into sentences or fragments [24] and preprocessed (e.g., to keep specific parts of speech only [25, 26]); features can then be extracted. General-domain ones include keywords [27], title words, cue words, and position. An anthology for technical terms might also be used [26, 28]. Domain-specific approaches [24] are often linguistically-motivated. Once extracted, features are used to score the sentence relevance in the summary either heuristically [27] or in a data-driven way [24, 29, 30, 31]. Alternative approaches use the patent discourse structure, which they prune [32].

Recently, [4] introduced the BigPatent dataset, whose associated task is that of summarizing the patent's Detailed Description into its Abstract. As authors show, patents' Abstracts are highly abstractive — with relevant content spread throughout the input — and have many novel n-grams. The dataset has been used as a testbed for general-purpose systems [19, 33, 34, 21, 35], given the high abstractivity of its targets and the length of its inputs.

For an overview of patent summarization approaches, see [36].

3. Dataset

We use the G (Physics) subsection of the BigPatent dataset [4].

The dataset is associated with the task of generating the patent Abstract from its Description. We are aware of the practical imitations of this setting, as the Abstract contains superficial and general information, but still consider experimenting on the dataset useful given its popularity in the Natural Language Processing community.

The dataset exists in two versions [3]. The original version text is uncased and tokenized, and its input typically contains the Detailed Description only (i.e., a subsection of the Description section). The alternative version contains the full Description with all its subsections, in the original casing. We will use this version in this paper. However, we notice two main limitations. First, patents lack section headers due to the performed preprocessing; thus, any structural information is lost. Second, the input often contains the author's Summary of the Invention, which significantly simplifies the task.

# docs		258,935
Summary	# tokens (avg)	121.0
	# sents (avg)	3.6
	sent len (avg)	43.4
Source	# tokens (avg)	4,893.6
	# sents (avg)	161.2
	sent length (avg)	31.3
compression ratio		45.8

Table 1
Length statistics on the BigPatent/G dataset. The number of tokens, sentences, tokens per sentence, and the compression ratio are computed per document and then averaged. The compression ratio is the ratio between the number of tokens in the source and the number of tokens in the Abstract.

To solve both issues, we download the raw data and (i) apply all original preprocessing steps, excluding removing the subsection headers and newlines, and (ii) remove the Summary of the Invention section by heuristically matching headers. Table 1 contains some metrics on our version of the dataset.

4. Evaluation Protocol

Evaluating patent summarization results is challenging.

On the one hand, automatic text simplification outputs (and Natural Language Generation outputs in general) are difficult to evaluate automatically, and the problem is considered open [37, 38].

While automatic metrics such as ROUGE [39] exist, they have known limitations. In the patent domain in particular, some previous work [40, 31] has anecdotally questioned the metric validity (and its correlation to expert's opinion and practical utility), even if no quantitative studies in the patent domain have been performed, to the best of our knowledge. More complex metrics, e.g., model-based methods [41, 42], should be fine-tuned with domain-specific data.

On the other hans, human evaluation is not easier. In fact, it is particularly hard in the patent domain for two main reasons: a) the best way to evaluate a summarization output is to read the whole source document. However, patents are extremely long and hard to read; b) patent documents and Abstracts are extremely complex and should be evaluated by legal and technical experts, but hiring such experts is very expensive and unpractical in most scenarios.

Aware of these limitations, we will use two main evaluation methods:

- Automatic evaluation: we will select hyperparameters and automatically evaluate outputs using ROUGE [39]. We also experimented with factuality-related metrics, e.g., QAEval [41]; however, they do not seem to adapt well to the patent domain and should be fine-tuned.
- Qualitative evaluation: we report a preliminary qualitative evaluation of a subset of candidate summaries. We will consider the patent fluency, consistency, and similarity to the Abstract.

5. Extractive methods

5.1. Graph-based systems

The core idea of graph-based methods is to represent the original document as a graph having sentences as nodes and their similarity as edges, and then extract the most central sentences only.

5.1.1. TextRank [7]

TextRank uses the number of shared words among two sentences, normalized by the length of the sentences as its similarity metrics. Edges in the complete graph are then pruned using a threshold, and the most central sentences according to PageRank [43] are extracted. We used the *summa*[1] implementation. In this implementation, the user chooses the target summary length in terms of tokens, and the number of sentences that best approximate that number is extracted. We cross-validated the number of tokens and left any other parameters at their default values. Some sample outputs are in Table **??**.

5.1.2. LexRank [6]

LexRank is similar in nature to TextRank, but it uses the cosine similarity of their Term Frequency–Inverse Document Frequency (TF-IDF) representation as its similarity metrics. We used the *sumy* implementation[2]. We validated the number of extracted sentences per patent and left any other parameters at their default value. The algorithms are unsupervised and can easily be used even for very long documents with no modifications. We also tried to perform experiments with PacSum [8] but found the algorithm extremely computationally demanding in our use case.

[1] https://summanlp.github.io/textrank/
[2] https://github.com/miso-belica/sumy

Set	#T.	ROUGE-1	ROUGE-2	ROUGE-L
Val	50	28.20	8.52	18.08
Val	100	37.06	11.40	21.99
Val	150	**38.60**	**12.33**	**22.33**
Val	250	35.39	12.27	20.69
Val	500	25.74	10.37	16.11
Val	1000	16.22	7.65	11.00
Test	150	38.59	12.30	22.33

Table 2
Results using TextRank. We selected the number of extracted tokens on the validation test and run the most promising model on the test set.

Set	#S	ROUGE-1	ROUGE-2	ROUGE-L
Val	1	26.03	8.12	17.40
Val	2	34.72	10.93	21.14
Val	3	37.48	12.02	**21.89**
Val	4	**37.76**	12.40	21.71
Val	5	36.92	**12.46**	21.16
Val	6	35.62	12.36	20.48
Test	4	37.76	12.46	21.76

Table 3
Results using LexRank. We selected the number of extracted sentences on the validation test and run the most promising model on the test set.

Automatic evaluation ROUGE scores are shown in Table 2 and 3. As expected, performance is similar for the two systems, with TextRank being marginally superior. Unsurprisingly, the best-performing systems are those that select a number of tokens or sentences similar to that of the gold standard.

Qualitative assessment The outputs obtained using the two algorithms are relatively similar. We notice that the sentence tokenization is not always perfect: for example, the extracted summary of patent US-2005152022-A1 contains the sentence *"The mixed color display [...] by the type of processes described in the aforementioned U.S. Pat. No."*, where the patent number has been incorrectly considered as a stand-alone sentence. This is in accordance with previous work [44, 45], which showed that general-domain Natural Language Processing resources tend to have suboptimal performance in the patent domain and should be adapted.

Moreover, sentences naturally contain references to other parts of the original text[3] e.g., *"as described below"* in US.2005152011-A1 or *"according to claim 1"* in US-9478115-B2.

We also notice that all the extracted sentences tend to be extremely long and naturally contain core and peripheral information (e.g., included in parenthesis). These

[3] Sentences also tend to contain numerical references to the figures, which are lost.

Set	#S	ROUGE-1	ROUGE-2	ROUGE-L
Val	1	20.09	4.38	13.54
Val	2	28.51	6.48	17.15
Val	3	32.37	7.70	18.43
Val	4	33.93	8.38	**18.80**
Val	5	**34.28**	8.78	18.70
Val	6	34.00	9.02	18.43
Val	7	33.30	9.14	18.05
Val	8	32.44	**9.20**	17.63
Test	5	34.26	8.72	18.66

Table 4
Result using LSA. We selected the number of extracted sentences (#S) on the validation test and run the most promising model on the test set.

are known limitations of naive extractive models and are very common problems of our extracted summaries. Extracted sentences do not seem too similar to each other, which is sometimes described as a limitation of graph-based systems.

Even with their limitations, the algorithms seem to perform reasonable content selection (with TextRank being superior to LexRank also from a qualitative perspective); when compared to their references, the extracted summaries often contain most of their core elements and, in many cases, are very similar to the reference in terms of content. This is evident in some specific cases (e.g., patent US-9478115-B2 and US-2003016244-A1) and is interesting, considering the algorithm is unsupervised.

If we assume the final target of the extracted summaries is human readers, the lack of discourse structure and the length of the extracted sentences might make the outputs too hard to understand. It might, however, be possible to use the outputs in an ad hoc interface, e.g., where core sentences are highlighted.

5.2. Latent Semantic Analysis

Latent Semantic Analysis [46] aims at exploiting the latent semantic structure of the document and extracts sentences that best represent the most important latent topics. The algorithm decomposes the term-sentence matrix constructed from the source document using SVD [47]. The $t \times s$ terms-by-sentence matrix A is thus decomposed as $A = U\Sigma V^T$. Thus, the original matrix is decomposed into a matrix of term distributions over latent topics, a diagonal matrix of topic importance (the singular values), and a matrix of topic distributions across sentences. For each of the K most salient latent topics (i.e., those corresponding to the largest singular values), the sentence with the largest index value is included in the summary [48, 10]. We use the *sumy* implementation, validate the number of sentences, and leave all other parameters at their default values. Some sample outputs are in Table **??**.

Set	ROUGE-1	ROUGE-2	ROUGE-L
Validation	41.70	17.52	28.38
Test	41.53	17.25	28.18

Table 5
RBART results on the validation and test sets.

Automatic evaluation Table 4 shows the ROUGE scores. LSA tends to perform worse than the graph-based algorithms. In contrast to the graph-based methods, it tends to work best when extracting several short sentences.

Qualitative assessment Even with the known limitations of extractive systems (references, structure, sentences needing compression, etc.), some reasonable content selection is performed. For example, they often extract the sentence that describes the invention's nature, as in *"The present invention is based on the object to provide an operator system for a machine, which is ergonomic with regard to the handling thereof and offers sufficient work protection."* for US-9478115-B2 or *"The present invention relates to computer security and, more particularly, to an efficient method of screening untrusted digital files."* for US-9208317-B2. Sentences are generally shorter than those extracted by graph-based systems.

[31] noticed LSA showed a better quality when compared to TextRank in the generation of patent titles. Our results do not confirm this finding for Abstract generation from the Description as measured automatically; qualitatively, the results are relatively different and might be used for different purposes.

6. Abstractive methods

We use BART [18], a sequence-to-sequence system, as a baseline for abstractive summarization. We fine-tune a BART-base model ($\sim$ 140 million parameters) on the BigPatent/G datasets. We train using the Hugging Face library with early stopping on the evaluation loss (patience: 5) and the following hyperparameters: max target length: 250; number of beams: 5; evaluation steps: 10k; max steps: 500M. We leave all other parameters at their default values. Some sample outputs are in Table **??**.

Automatic evaluation Table 5 shows the results in terms of ROUGE. As expected, the results improve over all extractive systems, with an increase of almost 5 ROUGE-2 points over the best extractive system.

Qualitative assessment Qualitatively, we notice that summaries are generally grammatical, with very rare local problems. Text is coherent and much easier to read and understand than those composed through extracted

	BART	Hybrid	Gold standard
Coverage (avg)	95.75	96.12	90.68
Density (avg)	11.84	8.83	3.82

Table 6
Extractivity metrics on the summaries generated by the fine-tuned BART and the select and rephrase models. We also report the corresponding metrics on the gold-standard summaries for comparison. The metrics are computed per document and then averaged.

Set	#T	ROUGE-1	ROUGE-2	ROUGE-L
Val	1000	**42.79**	**17.92**	**28.79**
Val	500	41.54	16.74	27.88
Val	250	40.33	15.60	27.01
Test	1000	42.47	17.74	28.59

Table 7
Result using the previously described hybrid approach. We selected the number of extracted tokens (#T) on the validation test and run the most promising model on the test set.

sentences. In all cases, summaries seem adequate and convey the main points of their gold standard counterparts.

However, we noticed that the generated summaries are largely extractive, with no or few modifications to sentences in the source. In the following example, the extractive fragments in the summary generated for patent US-2005152022-A1 and its source (Background of the Invention subsection) are underlined.

More specifically, in one aspect this invention relates to electro-optic displays with simplified backplanes, and methods for driving such displays. In another aspect, this invention relates to electro-optic displays in which multiple types of electro-optic units are used to improve the colors available from the displays. The present invention is especially, though not exclusively, intended for use in electrophoretic displays.

While some deletion is performed, most text is directly extracted from the source. To quantify how extractive the generated summaries are with respect to the source, we compute the coverage and the density of the generated summaries, following [49], which we report in Table 6. The extractive fragment coverage measures the proportion of tokens in the summary that is part of an extractive fragment; it roughly measures how much a summary vocabulary is derivative of a text. The density also takes into account the length of the extractive fragments: the higher the density, the better a summary can be described as a series of extractions. We notice that the generated summaries tend to have much longer abstractive fragments with respect to the gold standard.

7. Hybrid methods

7.1. Extractive to abstractive: select and rephrase

Results in the previous sections show graph-based extractive methods tend to be able to select central content but lack any discourse structure. Using BART solved some of these issues, but the model can only summarize the first part of the patent document, as its input length is limited to 1024 subtokens.

Thus, in this section, we explore a hybrid approach. We first select important sentences using an unsupervised graph-based algorithm and then rewrite the content using an abstractive system. Specifically, we use TextRank as it performed best among the considered extractive models. We considered three extracted lengths: 1000, 500, and 250 tokens. Then, we train a BART system to rephrase the selected sentences to generate the target summary: we use the selected sentences as the input and the original gold standard as the target and fine-tuned the model. Some sample outputs are in Table **??**.

Automatic evaluation Table 7 reports the ROUGE scores. Extracting 1000 tokens through TextRank and then rephrasing the summary using BART results in the highest ROUGE, surpassing the vanilla BART approach on all metrics. The obtained metrics are the highest among all the extractive and abstractive models we considered.

Note that, even for the approaches where a smaller number of tokens is extracted, relatively good performances are obtained. Extracting 500 tokens results in scores only marginally worse than those obtained by a BART model fed with the first 1024 subtokens. While results obtained by extracting 250 tokens only score worse in terms of ROUGE, the rewriting component is crucial. In fact, an improvement of 5 ROUGE-1, 3.3 ROUGE-2, and 5.3 ROUGE-L points is observed over the results obtained using TextRank only.

Qualitative assessment The outputs obtained with this approach are fluent, and relatively similar to those obtained through the vanilla BART. The coverage and density (Table 6) also show a marginally lower extractivity of the generated summaries.

7.2. DANCER

An alternative approach to deal with high document length is to exploit the document structure. To summarize scientific documents, for example, [5], proposed to deal with different sections independently; however, no experiments were performed in the patent domain.

Here, we explore if adapting this method to the patent domain can be useful.

Specifically, we perform the steps described in the following.

- Dividing and normalizing subsections: To divide the Description text into subsections, we use simple regular expressions, exploiting the fact that section headers lines include fully cased tokens only. Patent headers can follow different conventions[4]. Thus, we normalize the headers through a simple keyword-matching algorithm into nine classes. The classes are shown in Table 8. Subsections that did not match with any of the keywords were left in a default category and ignored.
- Alignment between abstract sentences and subsections: Following [5], we use ROUGE-L [39] to align sentences in the abstract to patent subsections. Specifically, for each sentence in the Abstract, we compute its ROUGE-L recall with all individual paragraphs in all subsections; we then align the sentence with the subsection containing the paragraph with the maximum score[5]. Figure 3 shows the percentage of subsections that, when present, align with at least one sentence in the patent's Abstract.
- Using paired elements as training data: Following the previous steps, each Abstract sentence is aligned with a Description subsection. Thus, for each (Description, Abstract)$_i$ pair, we created N (Subsection, Abstract sentence(s))$_{i_n}$ pairs, where N is the number of unique subsections that are aligned with at least one sentence in the Abstract. If multiple sentences align with the same patent subsection, the target contains all the aligned sentences in their original order. We then trained a BART-base model [18] using the subsection as input and the aligned sentence(s) as target; we set the maximum generated length to 250, the number of beams to 5, and left all other hyperparameters at their default values. We trained with early stopping on the validation set. Table 9 reports the metrics obtained by the model on the sentence generation step. We also experimented with prepending the subsection type (as a special token) to its text but with no improvement.
- Inference: At inference, we obtain the final summary by concatenating the sentences generated from the individual subsections. Patent structure is less coherent than that of papers; in fact, not all subsections appear in all patents. We thus consider several strategies for subsection selection:
  (i) Pre-selection: We heuristically pre-select subsections based on their role[6] and fed them to the trained model in their original order. We then concatenated the results.
  (ii) Generate from M subsections: We retrieve all subsections in the patent and sort them according to how likely they are to be aligned in the whole dataset (Figure 3). We generate from the first M most commonly aligned subsection, where M goes from 1 to the total number of subsections in the patent. The final summary is a concatenation of the generated sentences.
  (iii) Generate from all subsections in the patent: we use all subsections in their original order and concatenate the results.
- Second abstractive step: The final abstract obtained as a concatenation of sentences lacks any discourse structure and might not be coherent; in particular, we notice that it often contains repeated information. Thus, we explore if performing a second abstractive step can improve performance. To this end, we train a second BART model that, given the output of the previous step (i.e., the summary as a concatenation of sentences), is trained to paraphrase it to be more similar to the target Abstract.

Some sample outputs (before performing the second abstractive step) are in Table **??**.

	#Tokens	% patents
Field	73.73	38.27%
Background	710.04	94.85%
Drawings	243.43	97.60%
Embodiments	3168.25	53.07%
References	92.10	28.18%
Related Art	644.27	4.12%
Objective	256.95	2.09%
Detailed Descr.	3404.91	55.23%

Table 8
Average length of each subsection type and percentage of patents that contain the subsection.

Automatic evaluation Table 10 reports the results on the validation set. We report results obtained by generating from pre-selection, using the best-aligned section only (as a baseline), the best result with a varying number of sections (and Figure 2 shows ROUGE-L as a function of

[4]For example, subsections with similar content can be named Fields, Field, Field Of The Invention, etc.

[5]We retrieve the subsection containing most of the sentence content, regardless of any possible additional text (that the summarization model will learn to filter out).

[6]We selected the subsections of type FIELD, BACKGROUND, EMBODIMENTS, OBJECTIVE, DESCRIPTION

[name=plotLeft, scale=0.45, tick label style=font=, ylabel near ticks, xlabel style=yshift=2.2ex, xticklabel style=rotate=90, title=Train aligned sections, title style=yshift=-1.5ex,, symbolic x coords=BACKGROUND, DESCRIPTION, DRAWINGS, EMBODIMENTS, FIELD, OBJECTIVE, REFERENCES, RELATED ART, xtick=data,] [ybar] coordinates (BACKGROUND, 48.66) (DESCRIPTION, 87.02) (DRAWINGS, 1.76) (EMBODIMENTS, 83.49) (FIELD, 22.51) (OBJECTIVE, 2.14) (REFERENCES, 2.02) (RELATED ART, 33.65) ; [at=(plotLeft.right of south east), anchor=left of south west, scale=0.45, tick label style=font=, ylabel near ticks, xlabel style=yshift=2.2ex, xticklabel style=rotate=90, title=Val aligned sections, title style=yshift=-1.5ex,, symbolic x coords=BACKGROUND, DESCRIPTION, DRAWINGS, EMBODIMENTS, FIELD, OBJECTIVE, REFERENCES, RELATED ART, xtick=data,] [ybar] coordinates (BACKGROUND, 49.31) (DESCRIPTION, 86.67) (DRAWINGS, 1.86) (EMBODIMENTS, 83.59) (FIELD, 23.04) (OBJECTIVE, 29.21) (REFERENCES, 1.79) (RELATED ART, 34.74) ;

Figure 1: Percentage of subsections that, when present, are aligned to at least one sentence in the Abstract in the train (left) and validation (right) sets.

Model	R1	R2	RL
BART	35.00	15.74	26.63
BART$_{(+ \text{ subs. type})}$	33.28	14.81	25.66

Table 9
Model trained on generating the Abstract sentence(s) given the subsection. We also experimented with prepending the subsection text with its type.

Model	R1	R2	RL
DANCER $_{(\text{preselection})}$	38.73	16.03	25.63
DANCER $_{(\text{best aligned, M=1})}$	27.39	10.64	19.83
DANCER $_{(\text{best M, M=3})}$	40.70	16.45	25.08
DANCER $_{(\text{all})}$	40.68	16.38	25.90
DANCER + abstractive	38.88	15.89	26.99

Table 10
Results on the validation set.

[xlabel= Number of sections, ylabel= ROUGE-L, width=7.7cm,height=7cm] [color=black,mark=x] coordinates (1, 17.95) (2, 22.46) (3, 24.42) (4, 24.40) (5, 24.39) (6, 24.39) (7, 24.39) ;

Figure 2: ROUGE-L results as a function of the number of subsections used for the generation.

the number of summarized subsections), and the result obtained by summarizing all sections. We also report the results after the second abstractive step. Note that none of the configurations surpasses the simple BART baseline.

Qualitative analysis Inspecting the outputs, we noticed that many of the sentences generated from various

[height=5cm, width=7cm, ybar interval, ymin=0, ymax=150000, xmin=0.5,xmax=7.5, minor y tick num = 1, xlabel=distinct subsections, ylabel=Abs. # of Abstracts,] +[ybar interval, mark=no, draw=black, fill = white] plot coordinates (1, 130664) (2, 113373) (3, 13537) (4, 1196) (5, 138) (6, 24) (7, 3) ;

Figure 3: Number of unique subsections types to which the Abstract aligns.

subsections are very similar and describe what the invention is and its goal. While the second abstractive step helps limit repetition, the resulting output is often short and contains too little information compared to the gold standard. We noticed a number of issues that could make the transfer from the scientific publications to the patent domain unsuccessful:

- Less predictable structure and session headers: Scientific papers have a very coherent structure as they tend to roughly follow a fixed schema (e.g., Introduction, Previous Work, Method, Conclusions), with each section having a clear fixed role. While, on a superficial level, patent documents have a similar structure with sections and subsections, they are less coherent. As Table 8 shows, the subsections of the Description tend to vary. Moreover, the role of each subsection is less determined.
- Less compositional Abstracts: An analysis of the Abstracts' compositionality shows that many of the sentences in the Abstract align with the same patent subsections. Figure 3 represents the number of unique sentences to which each Abstract aligns. Note most patent Abstracts only align to one or two different subsections. Moreover, a qualitative analysis of the Abstract shows that while paper abstracts tend to follow a fixed structure (first describing the background, then the goal and methods, then the results and conclusions), patent Abstracts seem to lack the compositional nature of scientific papers. The lack of a fixed flow in the Abstract might also explain the relatively low results obtained by the abstractive model when generating the Abstract sentence(s) from the original subsections. As the alignment is more random, finding a pattern and correctly generating the aligned sentences is more challenging.

8. Conclusions

In this paper, we have benchmarked several extractive, abstractive, and hybrid methods on the BigPatent/G dataset.

Among extractive systems, we found that graph-based ones seem appropriate for content selection and perform relatively well in metrics and outputs. However, the extracted outputs are subject to all the limitations of extractive summarization, with dangling references being particularly common. The length of the sentences, the dangling references, and the lack of discourse structure make the outputs challenging to process for humans and possibly machines.

Among the abstractive approaches, we have analyzed BART and have found that it performs best in automatic metrics compared to extractive algorithms. We have also found that the produced outputs are, in fact, not very abstractive with respect to the input, with long chunks of texts identical to input passages; the model seems, however, very good in removing non-central content from the single sentences, which extractive systems are natively unable to do. In future work, we plan to explore more powerful abstractive models, including those in the GPT family [50, 51, 52].

We have considered a simple select-and-rewrite approach, which obtained the best automatic metrics. We have also tried to adapt DANCER, initially designed for scientific articles, to the patent domain. However, we have found that patents are more variable in the sections they contain and in the sections' content itself, and their Abstracts tend to be less compositional than those of papers. Thus, the approach was not particularly successful when transferring to the patent domain.

Our setting, however, has several limitations. First, the BigPatent dataset has known issues, and the Abstract is not regarded as the best target for summarization in the patent community, as it contains superficial information rather than the core invention nature. Second, we did not have the opportunity to collaborate with legal and technical experts to evaluate our outputs.

We believe that future work on patent summarization should tackle a number of open problems. First, we hope that this work will motivate the creation of better benchmarks, which can be shared among researchers and practitioners interested in patent summarization. Second, we hope that the design of such a benchmark can be made in conjunction with patent experts and industrial practitioners to ensure that it can be practically useful; while it is likely not practical to ask experts to write gold-standard summaries, there is space for improvement in the current setting. Third, the validity of the standard evaluation metrics in the patent domains should be measured based on experts' evaluation of the outputs. Finally, the factual accuracy of abstractive methods — which is particularly important in a legal and technical domain —- should be better investigated.

Acknowledgments

We acknowledge the support of the PNRR project FAIR — Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU

References

[1] I. Mani, D. House, G. Klein, L. Hirschman, T. Firmin, B. Sundheim, The TIPSTER SUMMAC text summarization evaluation, in: Ninth Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Bergen, Norway, 1999, pp. 77–85. URL: https://aclanthology.org/E99-1011.

[2] T. Sakai, K. Sparck-Jones, Generic summaries for indexing in information retrieval, in: Proceedings of the 24th Annual International ACM simplificationR Conference on Research and Development in Information Retrieval, SIGIR '01, Association for Computing Machinery, New York, NY, USA, 2001, p. 190–198. URL: https://doi.org/10.1145/383952.383987. doi:10.1145/383952.383987.

[3] S. Casola, A. Lavelli, H. Saggion, What's in a (dataset's) name? The case of BigPatent, in: Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid), 2022, pp. 399–404. URL: https://aclanthology.org/2022.gem-1.34.

[4] E. Sharma, C. Li, L. Wang, BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 2204–2213. URL: https://www.aclweb.org/anthology/P19-1212. doi:10.18653/v1/P19-1212.

[5] A. Gidiotis, G. Tsoumakas, A divide-and-conquer approach to the summarization of long documents, IEEE/ACM Transactions on Audio, Speech, and Language Processing 28 (2020) 3029–3040. doi:10.1109/TASLP.2020.3037401.

[6] G. Erkan, D. R. Radev, LexRank: Graph-Based Lexical Centrality as Salience in Text Summarization, J. Artif. Int. Res. 22 (2004) 457–479.

[7] R. Mihalcea, P. Tarau, TextRank: Bringing Order into Text, in: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 404–411. URL: https://www.aclweb.org/anthology/W04-3252.

[8] H. Zheng, M. Lapata, Sentence centrality revisited for unsupervised summarization, in: Proceed-

ings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 6236–6247. URL: https://aclanthology.org/P19-1628. doi:10.18653/v1/P19-1628.

[9] A. Nenkova, L. Vanderwende, The impact of frequency on summarization, Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005 101 (2005).

[10] J. Steinberger, K. Jezek, et al., Using latent semantic analysis in text summarization and summary evaluation, Proc. ISIM 4 (2004) 8.

[11] Y. Liu, M. Lapata, Text summarization with pretrained encoders, ArXiv abs/1908.08345 (2019).

[12] M. Zhong, P. Liu, Y. Chen, D. Wang, X. Qiu, X. Huang, Extractive summarization as text matching, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 6197–6208. URL: https://aclanthology.org/2020.acl-main.552. doi:10.18653/v1/2020.acl-main.552.

[13] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to Sequence Learning with Neural Networks, in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, MIT Press, Cambridge, MA, USA, 2014, p. 3104–3112.

[14] A. M. Rush, S. Chopra, J. Weston, A neural attention model for abstractive sentence summarization, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 379–389. URL: https://aclanthology.org/D15-1044. doi:10.18653/v1/D15-1044.

[15] S. Chopra, M. Auli, A. M. Rush, Abstractive sentence summarization with attentive recurrent neural networks, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 93–98. URL: https://aclanthology.org/N16-1012. doi:10.18653/v1/N16-1012.

[16] R. Nallapati, B. Zhou, C. dos Santos, Ç. Gulçehre, B. Xiang, Abstractive text summarization using sequence-to-sequence RNNs and beyond, in: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 280–290. URL: https://aclanthology.org/K16-1028. doi:10.18653/v1/K16-1028.

[17] A. See, P. J. Liu, C. D. Manning, Get to the point: Summarization with pointer-generator networks, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1073–1083. URL: https://www.aclweb.org/anthology/P17-1099. doi:10.18653/v1/P17-1099.

[18] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. URL: https://aclanthology.org/2020.acl-main.703. doi:10.18653/v1/2020.acl-main.703.

[19] J. Zhang, Y. Zhao, M. Saleh, P. Liu, PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization, in: H. D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 11328–11339. URL: http://proceedings.mlr.press/v119/zhang20ae.html.

[20] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The long-document transformer, ArXiv abs/2004.05150 (2020).

[21] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, A. Ahmed, Big Bird: Transformers for Longer Sequences, in: H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 17283–17297. URL: https://proceedings.neurips.cc/paper/2020/file/c8512d142a2d849725f31a9a7a361ab9-Paper.pdf.

[22] S. Huang, R. Wang, Q. Xie, L. Li, Y. Liu, An extraction-abstraction hybrid approach for long document summarization, in: 2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC), 2019, pp. 1–6. doi:10.1109/BESC48373.2019.8962979.

[23] W. S. El-Kassas, C. R. Salama, A. A. Rafea, H. K. Mohamed, Automatic text summarization: A comprehensive survey, Expert Systems with Applications 165 (2021) 113679. URL: https://www.sciencedirect.com/science/article/pii/S0957417420305030. doi:https://doi.org/10.1016/j.eswa.2020.113679.

[24] J. Codina-Filbà, N. Bouayad-Agha, A. Burga, G. Casamayor, S. Mille, A. Müller, H. Saggion, L. Wanner, Using genre-specific features for patent summaries, Information Processing & Management 53 (2017) 151 – 174. URL: http://www.sciencedirect.com/science/article/pii/S0306457316302825. doi:https://doi.org/

10.1016/j.ipm.2016.07.002.

[25] A. Trappey, C. Trappey, B. H. Kao, Automated Patent Document Summarization for R&D Intellectual Property Management, 2006 10th International Conference on Computer Supported Cooperative Work in Design (2006) 1–6.

[26] A. J. C. Trappey, C. V. Trappey, C.-Y. Wu, A Semantic Based Approach for Automatic Patent Document Summarization, in: R. Curran, S.-Y. Chou, A. Trappey (Eds.), Collaborative Product and Service Life Cycle Management for a Sustainable World, Springer London, London, 2008, pp. 485–494.

[27] Y.-H. Tseng, C.-J. Lin, Y.-I. Lin, Text mining techniques for patent analysis, Information Processing & Management 43 (2007) 1216 – 1247. URL: http://www.sciencedirect.com/science/article/pii/S0306457306002020. doi:https://doi.org/10.1016/j.ipm.2006.11.011, patent Processing.

[28] A. Trappey, C. Trappey, C.-Y. Wu, Automatic patent document summarization for collaborative knowledge systems and services, Journal of Systems Science and Systems Engineering 18 (2009) 71–94. doi:10.1007/s11518-009-5100-7.

[29] K. Girthana, S. Swamynathan, Query Oriented Extractive-Abstractive Summarization System (QEASS), in: Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, CoDS-COMAD '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 301–305. URL: https://doi.org/10.1145/3297001.3297046. doi:10.1145/3297001.3297046.

[30] K. Girthana, S. Swamynathan, Query-Oriented Patent Document Summarization System (QPSS), in: M. Pant, T. K. Sharma, O. P. Verma, R. Singla, A. Sikander (Eds.), Soft Computing: Theories and Applications, Springer Singapore, Singapore, 2020, pp. 237–246.

[31] C. M. de Souza, M. E. Santos, M. R. G. Meireles, P. E. M. Almeida, Using Summarization Techniques on Patent Database Through Computational Intelligence, in: P. Moura Oliveira, P. Novais, L. P. Reis (Eds.), Progress in Artificial Intelligence, Springer International Publishing, 2019, pp. 508–519.

[32] N. Bouayad-Agha, G. Casamayor, G. Ferraro, S. Mille, V. Vidal, L. Wanner, Improving the comprehension of legal documentation: The case of patent claims, in: Proceedings of the International Conference on Artificial Intelligence and Law, 2009, pp. 78–87. doi:10.1145/1568234.1568244.

[33] J. Pilault, R. Li, S. Subramanian, C. Pal, On Extractive and Abstractive Neural Document Summarization with Transformer Language Models, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 9308–9319. URL: https://www.aclweb.org/anthology/2020.emnlp-main.748. doi:10.18653/v1/2020.emnlp-main.748.

[34] J. He, W. Kryściński, B. McCann, N. Rajani, C. Xiong, CTRLsum: Towards Generic Controllable Text Summarization, arXiv preprint arXiv:2012.04281 (2020).

[35] M. Guo, J. Ainslie, D. Uthus, S. Ontanon, J. Ni, Y.-H. Sung, Y. Yang, LongT5: Efficient text-to-text transformer for long sequences, in: Findings of the Association for Computational Linguistics: NAACL 2022, Association for Computational Linguistics, Seattle, United States, 2022, pp. 724–736. URL: https://aclanthology.org/2022.findings-naacl.55.

[36] S. Casola, A. Lavelli, Summarization, simplification, and generation: The case of patents, Expert Systems with Applications 205 (2022) 117627. URL: https://www.sciencedirect.com/science/article/pii/S0957417422009356. doi:https://doi.org/10.1016/j.eswa.2022.117627.

[37] A. Celikyilmaz, E. Clark, J. Gao, Evaluation of Text Generation: A Survey, 2020. arXiv:2006.14799.

[38] E. Lloret, L. Plaza, A. Aker, The Challenging Task of Summary Evaluation: An Overview, Lang. Resour. Eval. 52 (2018) 101–148. URL: https://doi.org/10.1007/s10579-017-9399-2. doi:10.1007/s10579-017-9399-2.

[39] C.-Y. Lin, ROUGE: a Package for Automatic Evaluation of Summaries, in: Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, Barcelona, Spain, 2004, pp. 74–81.

[40] J.-S. Lee, Controlling Patent Text Generation by Structural Metadata, Association for Computing Machinery, New York, NY, USA, 2020, p. 3241–3244. URL: https://doi.org/10.1145/3340531.3418503.

[41] A. Wang, K. Cho, M. Lewis, Asking and Answering Questions to Evaluate the Factual Consistency of Summaries, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 5008–5020. URL: https://www.aclweb.org/anthology/2020.acl-main.450. doi:10.18653/v1/2020.acl-main.450.

[42] A. Pu, H. W. Chung, A. P. Parikh, S. Gehrmann, T. Sellam, Learning compact metrics for MT, in: Proceedings of EMNLP, 2021.

[43] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking : Bringing order to the web, in: WWW 1999, 1999.

[44] A. Burga, J. Codina, G. Ferraro, H. Saggion, L. Wanner, The challenge of syntactic dependency parsing adaptation for the patent domain, in: ESSLLI-13 workshop on extrinsic parse improvement., 2013.

[45] L. Andersson, M. Lupu, A. Hanbury, Domain

Adaptation of General Natural Language Processing Tools for a Patent Claim Visualization System, in: M. Lupu, E. Kanoulas, F. Loizides (Eds.), Multidisciplinary Information Retrieval, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 70–82.

[46] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. A. Harshman, Indexing by latent semantic analysis, Journal of the Association for Information Science and Technology 41 (1990) 391–407. doi:10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

[47] V. Klema, A. Laub, The singular value decomposition: Its computation and some applications, IEEE Transactions on Automatic Control 25 (1980) 164–176. doi:10.1109/TAC.1980.1102314.

[48] Y. Gong, X. Liu, Generic text summarization using relevance measure and latent semantic analysis, in: Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2001.

[49] M. Grusky, M. Naaman, Y. Artzi, 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, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 708–719. URL: https://aclanthology.org/N18-1065. doi:10.18653/v1/N18-1065.

[50] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training, Technical Report, 2018.

[51] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language Models are Unsupervised Multitask Learners, Technical Report, 2019.

[52] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language Models are Few-Shot Learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.