PGT: a prompt based generative transformer for the patent domain

Dimitrios Christofidellis [1 2] **Antonio Berrios Torres** [3] **Ashish Dave** [4] **Manuel Roveri** [3] **Kristin Schmidt** [5] **Sarath Swaminathan** [5] **Hans Vandierendonck** [2] **Dmitry Zubarev** [5] **Matteo Manica** [1]

Abstract

Patents are a valuable source of knowledge, but drafting them is a time-consuming and expensive task. Methods that assist patent generation can provide a two-fold improvement as they can speed up the generation process and suggest to the inventor ideas and claims. Herein, influenced by recent advances in language modeling via multitask learning and prompt engineering, we present Patent Generative Transformer (PGT), a transformer-based language model trained to facilitate patent drafting. Specifically, the model supports three tasks: part-of-patent generation, text infilling, and patent coherence evaluation. PGT complements inventors and assures the fast and successful transition from their input to a coherent patent disclosure taking advantage of its multitasking nature. We show how the model outperforms a collection of task-specific baselines on relevant metrics. We further test the quality of the generated text via blind testing by subject matter experts. Finally, we explore a zero-shot extension of the model showing how to use PGT for generating domain-specific abstracts.

1. Introduction

A *patent* is an exclusive right granted for an invention. It is an essential tool to establish ownership rights and protect the owner's intellectual property. Drafting a patent application is challenging and time-consuming. The process involves several stakeholders and countless iterations to draft a patent using the appropriate jargon. Besides the time investment, the cost of the drafting process can exceed 15,000$, depending on the subject of the invention. For the above reasons, natural language processing (NLP) methods that focus on the patent domain (Krestel et al., 2021) have gained a lot of interest recently. Especially for the patent generation process, methods that can generate parts of a patent in an automated manner have also been proposed in the literature (Lee & Hsiang, 2020; Lee, 2020). One of the most general and notable approach of this category is the PatentTransformer (Lee & Hsiang, 2020). This is a GPT-2 (Radford et al., 2018) based model trained from scratch on patent data to generate part of the patents.

The latest advances in Language Modeling (LM) showed that multitask learning (Collobert & Weston, 2008; Raffel et al., 2020) has drastically improved performance across different NLP applications. Recently, natural language prompting (Raffel et al., 2020; Sanh et al., 2021) emerged as an effective approach for multitask learning in NLP by reformulating specific tasks in the format of a natural language query/prompt that expects a response formatted accordingly. Inspired by these seminal works, we present Patent Generative Transformer (PGT), a transformer-based multitask language model (LM) trained to facilitate the patent generation process. Specifically, we cast three tasks of interest for the patent generation process, namely: part-of-patent generation, text-infilling and patent coherence evaluator as text generative tasks using the proper prompts. We train all the tasks simultaneously in a multi-task setting. We examine our model's quality and actual usage in three ways. First, we compare the model with other baselines for each designated task. Second, to highlight PGT's capabilities in actual scenarios, we perform an evaluation study of our work by consulting subject matter experts. Lastly, we move to a zero-shot setting and investigate to what extent the model can generate abstracts by providing solely a set of related keywords. The later use-case underlines the general capabilities of the PGT which are not confined to the patent generation process but can be extended to the broader knowledge discovery and hypothesis generation area. To facilitate the use of PGT, we have made the model available at `https://github.com/GT4SD/gt4sd-core`.

[1]IBM Research, Rüschlikon, Switzerland [2]Queen's University Belfast, Belfast, Northern Ireland [3]Politecnico di Milano, Milan, Italy [4]IBM Research, Hursley, United Kingdom [5]IBM Research, Almaden, United States. Correspondence to: Dimitrios Christofidellis <dic@zurich.ibm.com>.

ICML workshop on Knowledge Retrieval and Language Models, Baltimore, Maryland, USA, 2022.

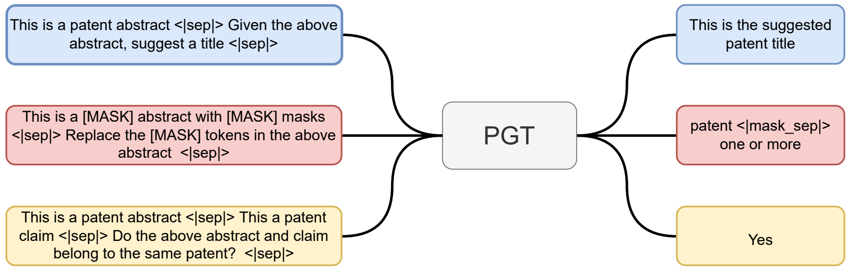


Figure 1: PGT model and prompt format. PGT is a GPT-2 based model that consumes inputs and produces target responses for three patent related tasks: part-of-patent generation, text infilling and patent coherence check.

2. Patent Generative Transformer

Automating the patent drafting process is the main goal of our work. Yet, we deviate from the existing approaches which focus solely on the task of part-of-patent generation and attempt to provide a multitask model that can perform a broader collection of useful tasks which can be used as tools during the patent generation. We call our model Patent Generative Transformer (PGT) and it is a multitask and prompt-based model relying on the GPT-2 architecture.

Being an autoregressive model, PGT is trained to predict the most suitable continuation of the given input. Working with a model that receives and produces text only, there is need to cast all the tasks of interest in a text format. Subsequently, the task identification is also done based on the input text by adding to it the respective task-specific prompt. The utilized prompts are small phrases that describe what is the input and what is the expected output, for instance “Given the above abstract suggest a title”. We place the task-specific prompt at the end of the input as this turns out to be the most effective strategy based on the literature (Reynolds & McDonell, 2021). The differentiation point for each task is prompt. The general approach is presented in Figure 1. For more information regarding the prompts, we refer the reader to the Appendix. PGT has been trained simultaneously on three tasks that could be utilized during the patent drafting process, namely: part-of-patent generation, text infilling and patent coherence check.

Part-of-patent generation is the standard generative language modeling task adapted to the patent domain. The model attempts to generate a part of a patent given as input another, already existing part of it. We focus on the title-to-abstract, abstract-to-claim transitions as well as their inverse. In the case of the claims, we focus on the first independent claim of the patents and at this point, we completely omit the introduction of the dependent claims in the task.

Patent editing is also desirable as the inventors of the patent may seek to improve a patent’s text or to enhance the patent’s quality by changing some of its parts. In this direction, the *text-infilling* task (Lewis et al., 2020) helps by suggesting alternatives for parts of the text that have been masked with the respective special token. The generated alternatives are not necessarily a single token but they could be a few words or even small phrases depending on the context of the input. To cast this task as a text generation problem, we reformulate the expected output. Specifically, the model’s output is a sequence of the suggested mask replacements in the same order as they are appeared in the input text and separated by a special mask separator token.

The previous tasks aim to generate or improve a part of the patent, however, there is also need to assess the quality and the coherence of the generated piece of text. To this end, the *coherence check* task treats this case as a binary classification problem and judges whether two given patent parts could belong in the same patent or not. This task is casted as a text generation problem by teaching the model to produce the output Yes or No depending on the coherence of the two inputs. Based on this format, we can also convert the output into a coherence score by inspecting the probability that the PGT model assigns to the Yes output.

A large patent dataset is mandatory to train such a model. In our case, we relied on a corpus of 11,600,000 patents that have been published between 1998-2020. Using all these patents, we generated 140,000,000 different instances that cover all the three tasks of interest and their different sub-cases. We refer the reader to the Appendix for further details regarding the dataset generation.

PGT has the standard GPT-2 architecture. We opted for GPT-2 instead of more recent models, such as T5 and GPT-3, due to their size. GPT-2 has at least one order of magnitude fewer parameters than the other two models, which makes its training feasible even with limited resources. Additionally, following a low budget approach and given that pre-trained GPT-2 has already demonstrated its power in general natural language, we preferred to not train our model from scratch (an approach followed in PatentTransfrormer) but to continue the training in a domain adaptive fashion (Gururangan et al., 2020). Specifically, we continued the training from the GPT-2 checkpoint using our patent dataset having a training budget time of one week which was translated

Table 1: Comparison of the performance of PGT and other baselines for the three tasks of interest. For each task, we examine all the different possible cases. Mean semantic similarity is used as a metric for the text-infilling and text generation cases while mean accuracy is used for the coherence check. Entries without values indicate that the model of this row could not be utilized for the respective use-case of this column. The best-achieved score for each case is highlighted in bold.

Model	text-infilling abstract	text-infilling claim	text generation title-to-abstract	text generation abstract-to-title	text generation abstract-to-claim	text generation claim-to-abstract	coherence check
BERT	0.37	0.40	-	-	-	-	**0.99**
BART	0.26	0.26	-	-	-	-	-
GPT2	-	-	0.36	0.42	0.36	0.47	-
PatentTransformer	-	-	0.54	0.65	0.64	0.66	-
GPT2-title2abstract	-	-	0.55	0.46	0.55	0.55	-
GPT2-abstract2title	-	-	0.52	0.55	0.42	0.37	-
GPT2-abstract2claim	-	-	0.52	0.51	**0.68**	0.62	-
GPT2-claim2abstract	-	-	0.53	0.55	0.53	**0.67**	-
PGT	**0.57**	**0.65**	**0.56**	**0.67**	0.66	**0.67**	**0.99**

into 3 full epochs. The system that was used for training contains 32 A100 GPUs.

3. Results

To assess the capabilities of our model, we proceeded with three different evaluations. First, we focused on the three tasks of interest for the model and we compared PGT's performance with different baselines. Secondly, we tested the quality of the generated text via blind testing by subject matter experts. Lastly, we assessed the zero-shot capabilities of the model in an unseen generation task to emphasize the model's understanding of the broad patent domain.

3.1. Evaluation of Tasks

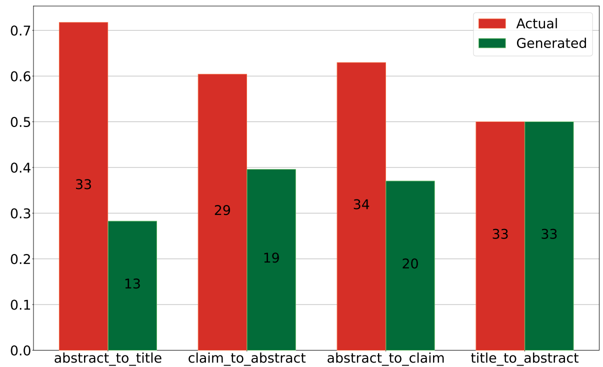
The first evaluation goal was to compare the PGT's abilities with other baselines for each task. Thus, we created an additional test set with 1000 patents. Using this set, we generated instances for each task following the procedures described above for the training set generation. For the part-of-patent generation, the aim was not to generate the actual part of the patent per se but to generate valid suggestions. For this reason, we avoided using word-based metrics and focused on the semantic similarity between the actual and the generated part to assess the models' quality, a metric that has been used extensively in previous studies (Lee & Hsiang, 2020; Lee, 2020). To measure the semantic similarity, we used the cosine similarity of the text embeddings extracted relying on sentence transformers (Reimers & Gurevych, 2019). The same motivation stands also for the text-infilling task. Also, we examined the semantic similarity between the actual and the generated masked content. Lastly, the patent coherence task is a binary classification problem, and therefore the accuracy of the models was examined in this case.

We compared our model with other baselines for each task. For the part-of-patent generation task, we used as baselines the standard pretrained GPT-2 model without further task-specific finetuning, PatentTransformer and 4 finetuned GPT-2 models - one for each of the subtasks of interest. To finetune these 4 models, we used the respective task instances from our training dataset and trained GPT-2 for 3 more epochs. For the text-infilling task, we used the pretrained BART (Lewis et al., 2020) and BERT (Devlin et al., 2019). Note that BERT is limited to replacing each mask with just one token. This limitation does not hold for BART. As it concerns the patent coherence check, we selected a BERT model fine-tuned for the task as a baseline. To train this model, we followed the standard BERT task-specific finetuning methodology (Devlin et al., 2019) and trained the model for 3 epochs.

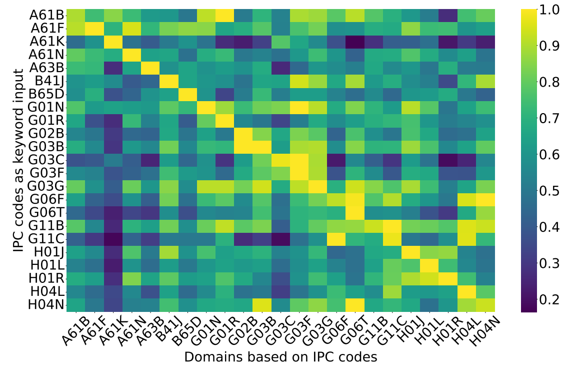
Table 1 depicts the mean performance of PGT and the other baselines for the three tasks of interest. Overall, the multi-task training helps PGT to have better or the same performance as the baselines. In the text-infilling task, PGT significantly outperforms the other baselines. In the part-of-patent generation, PGT is 2% better than PatentTransformer. This finding indicates that even though the two extra tasks are not directly relevant to the part-of-patent generation, they assist the model to understand better the patent domain. In some cases the case-specific finetuned GPT-2 model performs equally or slightly better results, yet the use of different models for each task/case would increase significantly the storage footprint of the patent drafting tool. Lastly, in the patent coherence check PGT and BERT classification both have the same high level of accuracy.

3.2. Evaluation by Subject Matter Experts

The above investigations focus on the quantitative comparison between PGT and other baselines. However, such evaluation cannot highlight the value of the model in real cases. To emphasize this, we performed a subject matter experts evaluation of the model focusing on the part-of-patent generation task. We focused on a small set of 44 patents related to the broad chemistry domain. For each of them, we used PGT and performed all the possible generative combi-



(a) Experts-based evaluation of PGT. Number of votes that the actual and the generated patents received from the experts for each of the 4 different generation cases.



(b) Evaluation of the zero-shot keyword-to-abstract generation. The similarity between the abstracts generated using as input keywords related to the respective IPC codes and actual patents from different domains grouped by their IPC codes. The values are normalized per row.

Figure 2: Evaluation of PGT capabilities.

nations of the part-of-patent task to perform part of patents. Then, we asked 8 patent experts to select given the input part which of the generated or the actual part of the patent suits better to the given part. The information regarding which is the actual and generated text was not exposed to the evaluators, so they selected solely based on the structure and the content of the two texts. Figure 2a presents the results. Interestingly, more than 37% of the selected texts are generated in the abstract-to-claim, claim-to-abstract and title-to-abstract cases. Our evaluation was confined to the part-of-patent generation task and there was not any human correction or any other postprocessing step to improve the generated texts before the evaluation, indicating that the generated texts were already in a good state and the needed intervention effort of a human to improve them would be typically low.

Evaluators' feedback contains also useful information for future directions in the field. Specifically, it has been highlighted that the model provides quite specific or detailed descriptions in many cases, which is not preferred in a patent document where a more general description could cover a broader invention range. This is mainly indicated in the results of abstract-to-title generation, where the majority of the generated titles were deemed too specific to be patent titles. Moreover, it was highlighted that the best starting point of an automated patent generation process is the abstract as it is probably the most descriptive part of the patent. Yet, the addition of further details, like keywords or a more detailed description of the invention as a starting point, could improve the process.

3.3. Zero-shot evaluation

One reason that LMs have gained a lot of attention recently, is that they depict great performance even in few-shot or zero-shot settings (Radford et al., 2018; Sanh et al., 2021). Towards this direction, we also examined PGT generation capabilities for an unseen task during training. Specifically, we attempted to generate patent abstracts based on keywords and check their similarity with existing abstracts. Keywords are too short, general and lack syntax. Thus, an invention cannot be described precisely using only them. Therefore, we examined whether PGT could understand the area of interest and suggest ideas, in the form of patent abstracts, related to this specific domain. Such attribute would set PGT as an idea/hypothesis generator method as well. We used the short descriptions of IPC codes (WIPO, 2022) (in the level of sections) as keywords and we generated 10 patents for each keyword. Then, we examined the semantic similarity of the generated abstracts and abstracts from our original testing set that hold various IPC codes. Figure 2b depicts the results. The generated abstracts are more similar to the abstracts that hold the respective or similar IPC code, which validates PGT's capabilities. We refer the reader to the Appendix for further analysis of this investigation.

4. Conclusion

Patent drafting is a tedious and time-consuming process. In the context of this work, we attempted to leverage state-of-the-art NLP to present PGT, an LM that can be utilized to automate and speed up this process. The results indicate that the model holds interesting capabilities and outperforms existing methods. Patents are a unique language domain with specific syntax, vocabulary and need for different levels of information across the different parts of the document. Thus, the modeling of this domain is a difficult task. Both the semantic analysis and the opinion of subject matter experts indicate that our model can generate accurate patent content and assist in the whole patent generation process. Furthermore, the model holds a strong understanding of the patent domain which that can be leveraged for tasks such as ideas generation in a zero-shot fashion. Such findings set PGT as a useful tool that can be utilized either by inventors or researchers in different steps of the discovery process.

References

Collobert, R. and Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 160–167, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390177. URL https://doi.org/10.1145/1390156.1390177.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Gururangan, S., Marasovic, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. Don't stop pretraining: Adapt language models to domains and tasks. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. R. (eds.), *ACL*, pp. 8342–8360. Association for Computational Linguistics, 2020. ISBN 978-1-952148-25-5. URL http://dblp.uni-trier.de/db/conf/acl/acl2020.html#GururanganMSLBD20.

Krestel, R., Chikkamath, R., Hewel, C., and Risch, J. A survey on deep learning for patent analysis. *World Patent Information*, 65:102035, 2021. ISSN 0172-2190. doi: https://doi.org/10.1016/j.wpi.2021.102035. URL https://www.sciencedirect.com/science/article/pii/S017221902100017X.

Lee, J.-S. *Controlling Patent Text Generation by Structural Metadata*, pp. 3241–3244. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450368599. URL https://doi.org/10.1145/3340531.3418503.

Lee, J.-S. and Hsiang, J. Patent claim generation by fine-tuning openai gpt-2. *World Patent Information*, 62:101983, 2020. ISSN 0172-2190. doi: https://doi.org/10.1016/j.wpi.2020.101983. URL https://www.sciencedirect.com/science/article/pii/S0172219019300766.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. 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*, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2018. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Reynolds, L. and McDonell, K. Prompt programming for large language models: Beyond the few-shot paradigm. 2 2021.

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T. L., Raja, A., et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

WIPO. Guide to the international patent classification, 2022.

Appendix

Prompts and output format

Figure 3 presents the format of a model's input. The green part (straight line) is the expected input to the model while the orange (dashed line) part is the output that the model learns to generate and append to the input. Furthermore, in Table 2, the specific prompts that have been used for the tasks of interest are presented. For the zero-shot keyword-to-abstract generation that has been investigated the prompt for the part-of-patent generation has been used as well where "keywords" are placed in the part of the patent placeholder that refers to the input text type.

Patent dataset

We relied on a corpus of 11,600,000 patents that have been published between 1998-2020 to generate our dataset. Using all these patents, we generated 140,000,000 different instances that cover all the three tasks of interest and their different sub-cases. Specifically, for the part-of-patent generation, we generated 4 different instances for each patent based on the generation combinations described above. For the text infilling task, we generated 2 instances for each patent(one for the abstract and one for the claim). We selected randomly the number of masks in each instance between 1 and 3. The number of words per mask follows a binomial distribution with n=5, p=0.3 and a shift of one. Lastly, we assured that the masked part of the instance is less than 25% of its total number of tokens. For the patent coherence task, we need both positive and negative examples. The positive examples were easily created by parts that already belong to the same patent. To generate negative examples, we used a positive example and we replaced one of the parts with the respective part of a randomly selected patent. For each patent, we created 1 positive and 1 negative example.

Zero-shot evaluation

Figure 4 presents the non-normalized version of the results presented in Figure 2b as well as the distribution of the similarity scores of the generated abstracts and abstracts that belong to the same domain as the input keyword. Even if the generated abstracts are more similar to abstracts originating from the same domain, we can observe differences in the achieved similarity between each case. This is attributed to the broad range of applications that each domain can cover. The same stands also for the achieved level of similarity which even in the best cases for each row usually lies between 0.35 and 0.40, values way less than the ones achieved in the actual versus generated part of patent comparisons. Furthermore, based on the similarity scores distribution that we examined, we show that the understanding of the patent domain is attributed to our training and it is not inherited from GPT-2.

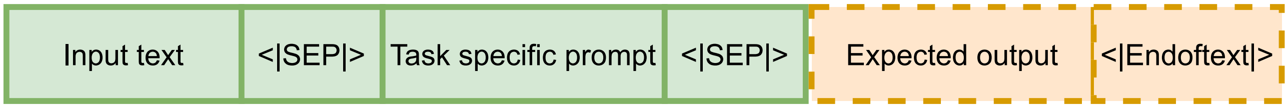
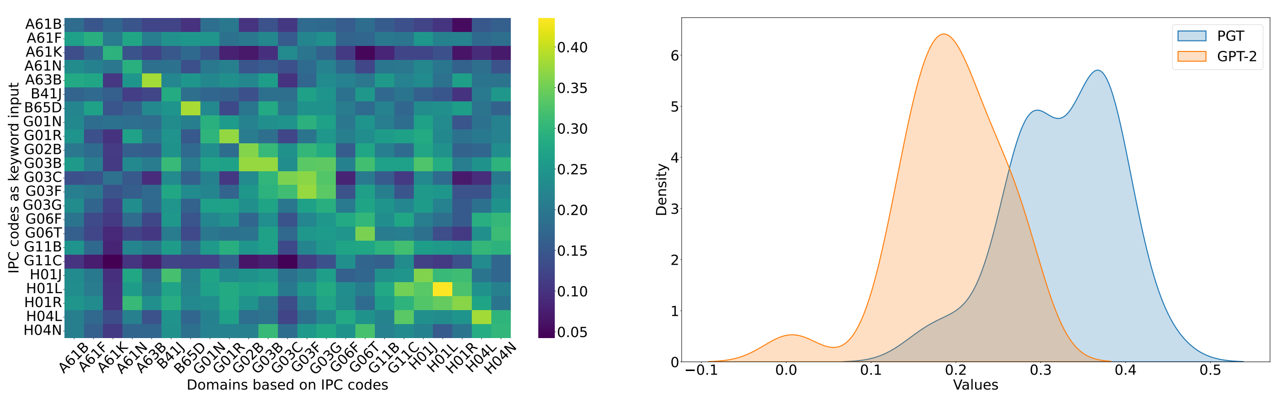


Figure 3: Prompt format used by PGT.

Table 2: Prompts and output format for each of the three tasks supported by PGT.

Task	Prompt	Expected output
Part-of-patent generation	Given the above <part of the patent> suggest a <part of the patent>	Generated part of the patent
Text infilling	Replace the [MASK] tokens in the above <part of the patent>	suggestion for 1^{st} mask $< \|mask\_sep\| >$ suggestion for 2^{nd} mask ....
Patent coherence check	Do the above <part of the patent>and <part of the patent>belong to the same patent?	Yes or No



(a) Similarity between the abstracts generated using as input keywords related to the respective IPC codes and actual patents from different domains grouped by their IPC codes.

(b) Distribution of the similarity scores between the generated abstracts and abstracts that belong to same domain as the input keyword using PGT or GPT-2 for the abstract generation.

Figure 4: Evaluation of the zero-shot keyword-to-abstract generation.