PatentEdits: Using LLMs to Rewrite Patents for Novelty

Anonymous ACL submission

Abstract

A patent must be deemed novel and non-obvious in order to be granted by the US Patent Office (USPTO). To meet this criteria, patent writers often revise the description of the claimed invention after official feedback is received.

In this work we examine how patents are revised to overcome objections to novelty. First, we present the PatentEdits dataset, the first to contain more than 400,00 granted patents aligned before and after revision. Next, we label the edit actions in our dataset: a given sentence in the patent is either *Unchanged*, *Edited*, or *Deleted*. We also include the prior work cited by the USPTO examiner during review and study how they influence the patent edits.

We explore a new research question for the community: how can language models learn to revise documents for originality? We demonstrate the promise of the following model pipeline for novelty revision: 1) the prediction of edit actions on the draft sentences using the prior work followed by 2) the prediction of the revised text with the edit actions.

1 Introduction

Patents are critical protections of a company's intellectual property and competitive advantage: they grant inventors the exclusive rights to make, use and sell the disclosed invention for 20 years.

Existing work such as the Harvard USPTO Patent Dataset (HUPD) (Suzgun et al., 2023) focuses on predicting the acceptance of a patent based on the first submission. However, by focusing solely on the first submission of a patent, HUPD is limited: 86% of all patent applications are initially rejected by the US Patent Office and then *revised* according to a 2015 Yale Law study (Carley et al., 2015). Given that most patents are not deemed novel and non-obvious based on first submission, we pose it is more critical to study how the initial patent is rewritten to understand acceptance.

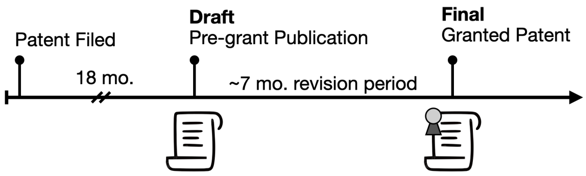

Figure 1: Simplified patent application timeline.

Moreover, we include and consider the USPTO feedback to the initial submission, which comes in the form of cited prior inventions which overlap and challenge the novelty of a patent.

In this work, we introduce *PatentEdits*, a dataset of over 483k patents and their corresponding cited references. With this dataset, we explore using large language models (LLMs) to revise documents for novelty, considering the overlapping prior work. We also investigate the best way to incorporate the overlapping prior work in order to improve generated revisions, and conduct ablation studies to test the usefulness of prior work.

To answer these questions, we systematically track the revisions to patent documents, following Spangher et al. (2022). After labeling patent document changes, we train models to predict those changes. We show how LLMs can revise for novelty by making the following contributions:

1. We introduce PatentEdits, the first bulk dataset which aligns drafts of patents with to final versions *and* examiner-cited prior work.

2. We track and analyze the revisions in PatentEdits, finding learnable edit patterns in the corpus, such as the fact that 66.6% of patents revise the first sentence.

3. We develop edit label classifiers and text editing models that can achieve a BLEU and Rouge-L score of 63.6 and 85.2 against the revised, successfully patented text.

2 Background

Why are patents edited? When a pre-existing invention is discovered to overlap with the claimed invention in the draft, the USPTO examiner notifies the inventor with a reference to the existing work. After this official feedback is given, the inventor must revise the patent application or risk losing the claims to the invention.

To overcome their patent being rejected for lack of novelty, patent writers will often add more detail and specificity *where there is overlap with a prior invention*. However, adding too much detail has drawbacks: the most valuable patent is one that is the most general and least descriptive, as it grants the inventor rights to any future invention that can be described in those terms. This leads to an incentive to change only what is necessary to establish semantic difference with related work. The sentences of the patent that provide legal coverage are called the claims, and the focus of this work. We study the evolution of these claims through the patent revision process.

3 The PatentEdits Dataset

PatentEdits consists of 483,076 patents and 1.3 million cited references from 2001 to 2014. Specifically, the dataset contains the patent claims text before and after revision as well as the claims text of cited references. This critical section of the patent describes the legal coverage of the invention claimed and is the primary focus during official review.

3.1 Dataset Collection

There exists no single bulk source containing both the draft and final claims as well as the cited references, so we extracted and aligned from 4 separate, publicly available USPTO datasets. In detail, utility patent claims text was extracted from two USPTO's Patent Claims Research Datasets (Marco et al., 2016), after which a third USPTO dataset, the Patent Examination Research Dataset (Graham et al., 2015) was used to align the initial claims text, called the pre-grant publication, to the final granted claims text. To obtain the list of examiner cited reference texts for each patent, the USPTO Office Action Citations Bulk Dataset was used. The dataset includes the USPTO classification of every patent example, however we do not filter the patents based on this for our experiments.

3.2 Examiner Cited References

PatentEdits also includes a set of cited reference documents (usually prior patents) provided by the US patent examiner during the official review of the draft claims. Although there are cited references from the patent writers themselves, we extract the subset cited by the US patent examiner, as these are the specific prior patents that must be worked-around with claim changes.

During patent prosecution, the examiner and patent writer may directly discuss the specific claims which must be changed, as well as the specific overlap in the prior patent cited; however, these exchanges are not readily available. As we detail later in Section 6, we can model this conversation by retrieving the most semantically similar sentences from the cited documents.

3.3 Edit Label Extraction

In order to study trends in patent revisions, we need to label which parts of the patent are edited, removed, added, or remain the same.

We define *edit actions* on draft sentences: *Edited*, *Deleted*, or *Unchanged*. We also track what final, granted sentences are *added*, or contributing new details not attributable to the draft sentences. Following Spangher et al. (2022), we first match draft sentences to the granted sentences, then as shown in Fig. 2, matched sentences are interpreted as edit actions by the following set of rules:

- *Unchanged*: a draft claim sentence is labeled as *unchanged* if a granted claim sentence exists that is identical.
- *Edited*: a draft claim sentence is labeled as *edited* if it is linked to at least one given grant claim sentence. This means the details in the draft claim were combined with others or added to in the final claim sentence.
- *Deleted*: a draft claim sentence is labeled as *deleted* if there is no granted claim sentence with sufficient similarity.

As detailed in Papineni et al. (2002), we perform the matching based on pair-wise sentence similarity, using BLEU-4 with equal weighting ($w_n = 0.25$ for $n = 1 : 4$) and smoothing of the n-gram precisions (p_n). First, we calculate the brevity penalty BP, where c and r is candidate and reference sentence length, respectively.

$$\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r \end{cases}$$

Then we calculate the BLEU match score with:

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right).$$

To attribute draft to final sentences, every draft sentence is attributed to the grant sentence it has the highest BLEU-4 score with. If the BLEU-4 score between draft and grant sentence is above 0.8, it is considered identical, and the draft sentence is labeled unchanged. If the highest score of a draft sentence is below a threshold of 0.2, it is labeled as deleted, as there is no existing grant sentence that is similar. All other pairs of sentences were labeled as edited. As described in section 3.4, we validate these thresholds by testing the automatic edit labeling against human evaluation. We note that PatentEdits also tracks the added, or new granted claims, however in this work we focus on what happens to the initial draft claims. Fig. 2 gives an example of the sentence matching algorithm algorithm and the extracted edit labels.

3.4 Edit Label Extraction Quality

To evaluate the quality of the automatic matching between draft and final sentences of the document, we instruct human annotators to match sentences between draft and final patent documents for 56 patents in the PatentEdits dataset. For all the sentences of the draft and final patent, annotators are instructed to match a given draft sentence to a granted sentence if they have substantial overlap in meaning, even in instances where the inventive detail has been paraphrased in the revised granted sentence. If there is no substantial conceptual overlap between the two sentences, or if it is unclear how the two sentences are semantically related, we instruct the annotator not to match the sentences.

To evaluate the quality of the edit labels assigned to the draft sentences with the algorithm, we obtain edit labels from the annotated sentence matches in the same manner we do with the algorithmically obtained matches. If a draft sentence has no annotated match, the draft sentence is labeled as deleted. If a draft sentence is matched to a grant sentence, that draft sentence is labeled to be unchanged if the BLEU match score is above 0.8, else it is considered edited[1]. We report the F1 classification score on the sentence matches with the human annotated labels as the ground truth in Table 1.

The sentence matching algorithm achieves an F1 score of 86.3 against human annotations. This F1 score is comparable to the F1 score of 89.1 seen in NewsEdits (Spangher et al., 2022) with BLEU-3 based matching algorithm on revisions of news articles. Although there is no comparable F1 scores for the downstream edit labels provided in NewsEdits, we observe good agreement with human evaluation for the *Unchanged* (96.8) and *Edited* (87.1) PatentEdits classes.

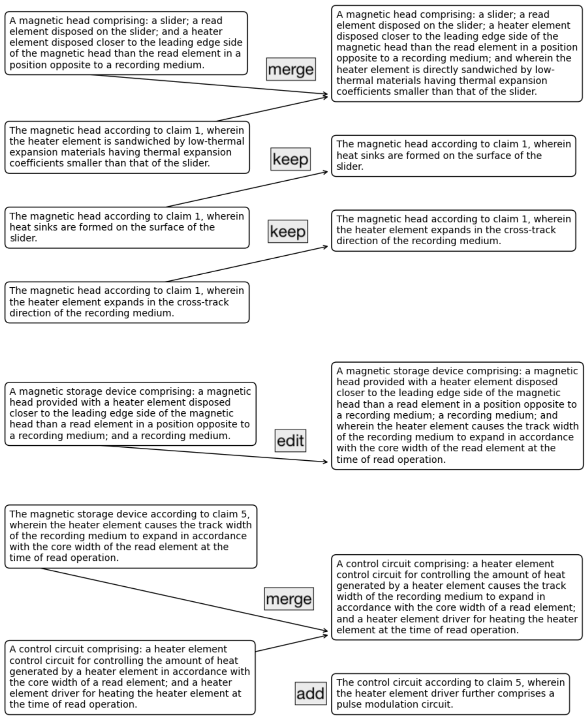

Figure 2: Shown are the extracted edit labels for US Patent 7561362. On the left are draft claims and on the right are granted claims, with edges denoting a sentence match. Additions are tracked but not yet considered.

	# Ann.	# Auto	F1
Matches	796	837	86.3
Unchanged	398	419	96.8
Edited	398	418	87.1
Deleted	160	119	60.9

Table 1: Edit extraction quality of automatic matches and labels against annotations, using F1 score per class.

[1]We note that this is a limitation that can be addressed in further studies: we could judge the difference between unchanged and edited with human evaluation vs. relying on an automatic threshold.

4 PatentEdits Dataset Analysis

Having successfully performed sentence-matching, we provide the following exploratory insights into how patents evolve after revision:

Insight #1: More draft claims will be unchanged than will be edited or deleted. As shown in Table 2, an average of 55% of a patent's sentences in PatentEdits are unchanged, resulting in a large degree of overlap between the draft and final claims. As seen in Table 3, this trend is reflected in the corpus-level measurement as well, with 52.5% of all sentences being labeled as unchanged.

This observation indicates that patent writers may be biased to make minimal changes to their patents following USPTO feedback and write focused, narrow revisions based on the overlapping inventive details brought to their attention by the examiner. This has implications for both edit-prediction and revised text prediction with LLMs: when most sentences are kept the same, locating where the revision occurs becomes an crucial preliminary step when predicting the final granted claims.

Insight #2: Two-thirds of the patents in the dataset have a first draft sentence that is edited. We observe in Fig. 3 the corpus-level edit label trends. At sentence index 0, or the first sentence, we observe that 66% of all patents in the corpus are labeled as *Edited*, and that after the third sentence, there is no increase or decrease of the relative percent of sentences labeled as edited. We also see that as the sentence index increases, it is more likely to see a higher percent of deleted sentences, while the number of unchanged sentences starts to decrease.

The insight that most patents have an edited first sentence matches our understanding of patent detail structure, in which the broadest, vaguest claims are made first with more specificity following later. It makes sense then that the first claim, with the broadest details, are often the ones which overlap the most with prior work, and need to be rewritten in order to seem novel and non-obvious to an examiner.

Insight #3: Most patent applications will have 1 or 2 cited references, but others have many. As shown in Fig. 4, we can observe by analyzing the number of references cited in the USPTO feedback that a majority of patents are found to overlap with 1 or 2 prior inventions. This means that in order to overcome the initial feedback, the patent writer must (1) identify the offending claim language that overlaps with the prior work, and (2) rewrite the draft patent in order to differ from the cited prior work. Since there are only a few prior cited works, only a few documents need to be used and referred to when rewriting the draft document in order to overcome the objection.

	Mean	Std. Dev
Unchanged	52.1%	±31.3%
Edited	31.7%	±25.4%
Deleted	16.2%	22.8%

Table 2: Per document statistics for each class in PatentEdits. Most claims are unchanged, but there is a lot of variation from patent to patent.

	Total Sents.	% of Sents.
Unchanged	4.7 mil.	52.5%
Edited	2.7 mil.	30.2%
Deleted	1.6 mil.	17.3%

Table 3: Summary statistics for each class in PatentEdits

5 PatentEdits Revision Pipeline

The dataset insights that roughly half of the draft claims are unchanged and the presence of clear edit patterns motivates us to take an approach for re-writing that involves sentence tagging followed by text generation, similar to work by Malmi et al. (2022) in LASERTAGGER. More specifically, we control LLM text generations by first predicting sentence-level tags that correspond to our edit labels (*Unchanged*, *Edited*, or *Deleted*) then selectively changing the sentences that are predicted to be *Edited* or *Deleted*.

By taking the preliminary step to identify which sentences should be *unchanged* vs. *edited*, we can preserve the value of the patent by, like a human patent agent, writing focused and narrow revisions. Concretely, we limit text generation to the subset of sentences predicted as *Edited*, thus training language models to rewrite tagged sentences as patent writers do.

As shown in Fig. 5, we define a pipeline where the edit actions on the draft sentences are first learned, via supervised learning with the automatically extracted edit labels, then the edited text is predicted using the edit action hypotheses. Two separate models are evaluated in this work for two separate tasks in the the pipeline: an *Edit Classifier* and a *Text Editor*.

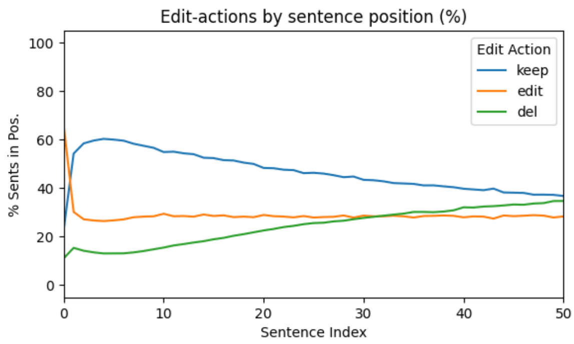

Figure 3: The first draft sentence at index 0 is the most likely to be edited. Deletions are more likely for later claims in the document.

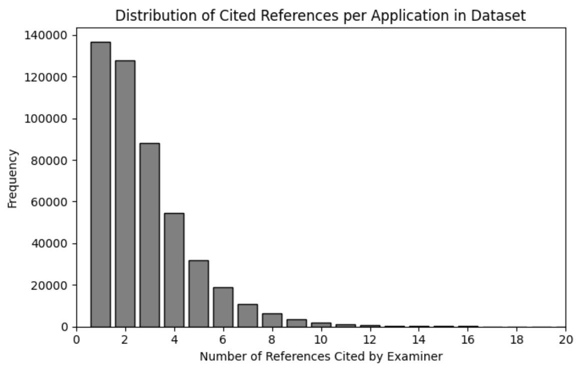

Figure 4: Most patent applications will have 1 or 2 examiner-cited references, but others have many. We leverage semantic search to find the most relevant reference on downstream tasks.

6 Patent Edit Classifier

In this section we explore how the patent edit labels can be predicted with text classifiers. For these experiments and for the text editing in the next section, we use a 10k random subset of the full dataset, with an 80/10/10 train/validation/test split. We also filter out for patents that have completely been rewritten and patents that were not revised at all. The experiments in this section are intended to illustrate how the included cited references and edit labels can be leveraged, and suggest the importance of the cited reference to the edit prediction task.

6.1 Edit Classification Experiments

Sentence-only Edit Prediction Given only the context of the draft sentence, we seek to predict the most likely the edit action that occurs on the draft sentence. Here, we investigate whether it is possible to predict the edit action given only the claim sentence, such as by identifying vague conceptual language that could not possibly be novel.

	Kept	Edit	Del
Sent	55.4	46.1	24.5
Sent+Cit	55.2	**49.6**	23.3

Table 4: F1-scores for each edit action are reported. We use RoBERTa-base for classification.

Sentence+Citation Edit Prediction Given both the context of the draft sentence and the most semantically similar cited reference sentences, we predict the edit action on the draft sentence. This experiment explores the signal that examiner-cited references have in influencing the edit outcome. As a pre-processing step we align the top-k most relevant sentences in the cited documents to each draft sentence using semantic search with sentence embeddings. We then simply concatenate the top 2 reference sentences for each draft sentences to that sentence for input.

6.2 Experimental Set-up

For these sentence-level prediction tasks, we utilized the RoBERTa-base architecture (Liu et al., 2019), a pre-trained BERT-based language model. For both tasks, we utilize under-sampling of the majority class on the training dataset to ensure that predictions for all classes are learned.

We separately fine-tune two RoBERTa-base models for edit classification, one trained on draft sentences with 2 reference sentences and one without the references. For both models we use the same batch size of 32, 6 training epochs, and a learning rate of 2e-5, with 500 steps of warm-up. Note these models only have the context from a single draft sentence and sentences are shuffled across patents during training.

For the **Sentence+Citation Edit Prediction**, we leverage neural retrieval models, similar to those outlined in Sentence-BERT (Reimers and Gurevych, 2019) to retrieve the top-k most relevant sentences in the cited reference documents for each draft sentence in the dataset. Specifically we use *gte-large-en-v1.5* (Li et al., 2023) a BERT-like encoder pre-trained on QA tasks and semantic search. In general, we found that semantic similarity searches worked better than automatic metrics due to the presence of many paraphrases of the same inventive concepts.

6.3 Edit Classification Results

As shown in Table 4, the Roberta-base edit classifier given the context of the draft sentence and

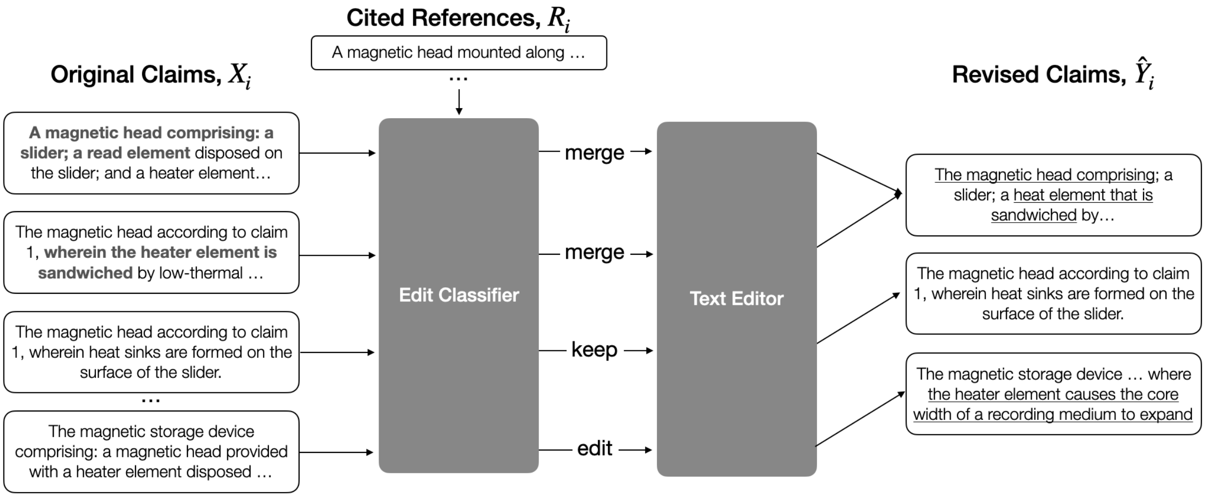

Figure 5: Our proposed pipeline with PatentEdits: we develop edit classifiers with Tasks 1 and explore text editing approaches in Task 2.

most semantically similar cited reference sentences is able to predict the *Edited* class with a greater F1 score than the classifier only given the context of the draft sentence. We do not observe that the additional context of the cited references improves the prediction of *Unchanged* or *Deleted* classes.

	BLEU-4	R-1	R-2	R-L
Mistral-7B	23.4	61.3	41.0	56.2
Llama-3-8B	24.4	62.0	41.5	57.4
GPT4	45.3	73.0	59.6	69.2
BART (FT)	**53.5**	**77.5**	**65.9**	**75.6**

Table 5: Sentence level results on *Edited* labeled sentences only. GPT4 performs the best of the in-context models. Fine-tuned BART outperforms all in-context baselines.

	BLEU-4	Rouge-L	BERT
LongT5	55.4	81.7	72.0
GPT4 w. Edits	60.0	83.5	77.8
BART w. Edits	63.6	85.2	**79.0**
Draft Doc.	64.5	86.1	**79.0**

Table 6: Document level test results. BART and GPT4 edit 33% of the total sentences in test. Despite the editing, we see that fine-tuned BART matches the semantic similarity of the draft document to the final claims

7 Patent Text Editor

In this section, we develop a text editor, which takes predicted edit actions and uses them to selectively revise the patent. By using edit actions as a guide, a model can learn to selectively edit a few sentences while keeping the majority of them, similar to how human writers are minimally revising patents after examiner feedback.

7.1 Text Editing Experiments

Revision without Edit Context Given the entire set of draft claims, we predict or generate the entire set of revised claims without the use of the extracted edit labels. We define this experiment to better understand whether the logic for editing vs. keeping can be learned implicitly by an attention-based transformer model.

Revision with Edit Context Lastly, we define the next experiment as follows: given a single draft claim of the patent (multiple in the case of merges) and the edit label, we predict and generate the revised granted claim. By revising a sentence at a time and excluding the context of the surrounding draft sentences, we explore how predictive the edit label context is alone. As a preliminary study, we take the edit labels to be perfectly found, to understand the upper bound of performance of a text editor in the PatentEdits revision pipeline.

7.2 Experimental Set-up and Models

For text revision without edit context, we utilize long-context models such as LongT5 (Guo et al., 2022) with efficient attention mechanisms. These efficient transformers enable us to process the entirety of the 800-1200 word patent draft without truncation and also generate longer outputs. We fine-tune LongT5 for 3 epochs, with Top-p sampling of 0.9 and temperature of 1.1 on the training dataset, and report the results in Table 6.

For text revision with known edit context, we consider two approaches: (1) rewriting *Edited* sentences using pre-trained LLM such as GPT4 with in-context learning and (2) fine-tuning BART (Lewis et al., 2020) on sentences labeled as *Edited*.

To choose representative examples for in-context learning, we leverage sentence embeddings as outlined in 6.2 to retrieve the top-k (k=2) most similar draft claims in the training set that have the same edit action. We then construct a prompt that includes the most relevant examples with the same edit action, as well as the given draft claim and edit action. We utilize GPT4, Llama-3-8B-Instruct, Mistral-7B-Instruct for these experiments, which are considered baselines against fine-tuned models.

We fine-tune BART (Lewis et al., 2020) to specifically rewrite the sentences labeled as *Edited* sentences while keeping or deleting the others according to the PatentEdits labels. BART was fine-tuned on the train dataset for 5 epochs, with a fixed learning rate of 5e-5 with an Adam optimizer and decoded with Top-p of 0.9 and a temperature of 1.1.

We score this approach on a sentence and document level: we first compare predicted edit sentences to the actual edit sentences as shown in Table 5 then compare at a document level by aggregating all the machine edited and unchanged sentences back into the full patent. At a document level, we also include the similarity scores of the initial draft patent claims (Draft Doc. in Table 6)

7.3 Text Editing Results

On a sentence revision level, fine-tuned BART outperforms our baseline model GPT4 with in-context learning with significantly higher BLEU (53.5) and Rouge-L (75.6) scores. Of the in-context learning models, GPT-4 outperforms our other baselines Mistral-7B and Llama-3-8B. Our results demonstrate that even given only the context of a single sentence and the edit action, we can reasonably predict the revised and granted claims. In Fig. 6, we show an example of a generated revision vs. an actual one.

On a document revision level, we observe that models that utilize the edit labels, such as fine-tuned BART, outperform Long-T5 which did not use the edit-labels. As most sentences are kept, it is likely that models without sentence-edit predictions (oracle labels in our initial experiment) will rewrite sentences that should be kept.

33% of the total sentences in the test set were labelled as edited and 13% are deleted. Thus, for

```
1. An internal combustion engine compris
ing: a first injector for injecting fuel
into an intake port or a combustion cham
ber; a second injector for injecting the
fuel into the combustion chamber followi
ng the injection of the fuel by the firs
t injector; and a spark plug for ignitin
g an air-fuel mixture within the combust
ion chamber, wherein an air-fuel ratio o
f the air-fuel mixture produced in the c
ombustion chamber by the injection of th
e fuel by the first injector is set in a
range of 28 to 38, and when a demanded o
perating load is changed, a ratio betwee
n an amount of gas residing in a cylinde
r and an amount of gas newly drawn there
in is controlled based on a closing timi
ng of an exhaust value.
```

```
1. An internal combustion engine compris
ing: a first injector for injecting fuel
into an intake port or a combustion cham
ber; a second injector for injecting the
fuel into the combustion chamber followi
ng the injection of the fuel by the firs
t injector; and a spark plug for ignitin
g an air-fuel mixture within the combust
ion chamber, wherein an air-fuel ratio o
f the air-fuel mixture produced in the c
ombustion chamber by the injection of th
e fuel by the first injector is set in a
range of 28 to 38, wherein an amount of
the fuel injected by the second injector
is fixed at a given value, and an amount
of the fuel injected by the first inject
or is changed corresponding to a demande
d operating load, and wherein when the d
emanded operating load is changed, a rat
io between an amount of gas residing in
a cylinder and an amount of gas newly dr
awn therein is controlled based on a clo
sing timing of an exhaust value.
```

Figure 6: A merged claim generated by BART on the left vs. the actual merged claim on the right. We note that more detail is added in the real merged claim, however there is high n-gram overlap between predicted and actual. Visual with DiffChecker (Diffchecker, 2023)

the BART and GPT4 document level predictions, at least a third of the sentences are changed. Despite the significant changes, we see that BART with the edit labels approaches the semantic similarity of the draft claims with the final claims, as measured with BERTScore (Zhang et al., 2020). We consider this a promising result for this edit based revision approach: we see a high semantic similarity with the final claims on par with the initial claims, all while prompting changes to a third of the draft sentences.

8 Discussion

With the PatentEdits edit classification and text editing experiments, we sought to explore how useful the edit labels and cited references are to understanding and predicting revisions for novelty. In our dataset analysis, we observe that the first, broadest patent claim, is often revised, but that roughly half of the remaining claims are untouched during revisions. These highlight the narrow and selective nature of the revisions, which initially motivated the edit-then-rewrite pipeline, explored in the experiments.

In the classification experiments, including the cited references improved predictions on which sentences would be edited: we interpret this as meaning that the cited references provide critical information on how the patent should be rewritten. In these experiments, the model was only provided the context of a single draft sentence and at most 2 additional cited reference sentences, but including more context, such as the whole draft and cited reference documents could further improve the classification performance.

In the text editing experiments, we compare a direct, sequence-to-sequence approach to text editing with no edit labels to an approach that uses known oracle edit labels and find that using the edit labels significantly increases the similarity of the text generation to the true granted claims. This is expected, given what we learned in the dataset analysis. As most draft patent sentences are unchanged, a model that can identify the narrow subset of sentences that should be changed and then revise them would be a better approximation of a human patent writer, who would do the same in the patent application cycle. We note that although we, either by prompt or fine-tuning, inject change into 33% of the draft claims (those labeled as *Edited*), we still achieve strong performance against the granted claims according to BLEU, Rouge, and BERTScore.

We note that using oracle-labels means that this is an “upper bound" on performance: further work that uses predicted, or silver edit labels instead of oracle labels would further confirm the efficacy of the pipeline outlined in this work. However, we note that the experiments outlined in this paper used only single sentence context for prediction and for text revision, which may be limiting the capabilities of the models employed.

9 Related Work

Pre-existing patent datasets for machine learning such as the Harvard USPTO Patent Dataset (Suzgun et al., 2023) focus on classifying initial patentability, or predicting patent field class. In contrast, we build PatentEdits to understand how the patent writer overcomes the prior cited references by revising their patent claims.

Lee and Hsiang (2020) described fine-tuning GPT-2 for claim generation. In our approach, we bring new focus towards using the sentence-level edit as prompt context and retrieve relevant edit examples using PatentEdits as a database.

The definition and extraction of sentence-level edit labels extends upon the work of Spangher et al. (2022) in the News domain: we adapt these methodologies for the patent domain by focusing on using examiner cited references to predict edits and focusing on predicting the revised patent claims text. The concept of using edit tags to guide generation is similar to the approach outlined by Malmi et al. (2019) with their LASERTAGGER model, however we define the use of sentence-level edit labels to guide generation at the sentence level, rather than word level.

10 Conclusions

We introduce the first bulk dataset that aligns the claims text data of patents before and after revision. Given the data insight that most draft sentences are kept, we demonstrate that PatentEdits can be leveraged to build a model pipeline that first predicts edit actions and selectively revises sentences. In this work we also provide experiments which explore the most effective approaches for predicting edits with the cited references and draft sentences. Finally, we demonstrate the importance of edit labels by showing how using the labels to selectively revise can significantly improve the prediction of revisions.

11 Ethical Considerations

11.1 Limitations and Risks

The edit actions in PatentEdits are determined based on rules and automatic metrics and verified with human evaluation. While the annotators were able to manually verify truthfulness for a subset of examples, the quality and correctness of the automatic may further improve with expert evaluation, i.e. by patent agents or USPTO examiners. In a similar vein, further expert evaluation would further verify the quality of the text editing by fine-tuned BART and GPT4.

Another limitation of this work is that we do not consider predicting the “added" claims. Although the PatentEdits dataset identifies these added grant claims, we do not define any edit prediction for added claims, as other works such as NewsEdits or LASERTAGGER do, i.e. whether a claim will be added before or after a given draft claim. We note that predicting the text of added grant claims (which do not have details in common with the draft claims) may require context from beyond the claims text section of the patent.

Another key limitation of the sentence-level approaches chosen for revision prediction is the ability to replicate the unique format and structure of the patent itself: specifically the aspect that sentences in a patent will refer and extend off of other sentences, i.e. “wherein the golf glove of claim 1 further comprises of a velcro fastener." Specifically, for the sentence level edits from GPT4 are simply concatenated together for the document level comparisons. However, a true patent would ensure the correct dependencies between sentences: although we did not take this step, this re-formatting may be achievable with a post-processing model or algorithm.

11.2 Privacy and Risks

We do not believe there to be any significant privacy risks associated with this dataset as patents are a matter of public record, and PatentEdits is aggregated from bulk datasets shared by the USPTO for the express purpose of research into the patent prosecution process. Although the USPTO Office Action dataset does contain personal identifiers for patent agents and examiners, only the examiner cited references were collected from that data source.

11.3 Computational Resources and Libraries

The PatentEdits dataset was processed with a TPU from Google Colab with 334GB of memory as well as with Google BigQuery. We share the processing code to obtain the PatentEdits dataset from the original sources, however extracting from scratch will require those resources. The fine-tuning experiments in this work are conducted using a NVIDIA V100 GPU with 40GB of GPU memory. The use of GPT4 requires OpenAI credits, and a total of $25 was expended for experiments and predictions with prompting.

We use HuggingFace libraries and models in this work, such as RoBERTa for edit prediction and encoders sentence-BERT from the Transformers library for extracting most similar edit examples as well as cited references. For evaluation, we utilize publicly available NLP libraries such as NLTK, scikit-learn, bert-score and rouge.

References

Michael Carley, Deepak Hegde, and Alan Marco. 2015. What is the probability of receiving a u.s. patent? Yale Journal of Law and Technology.

Diffchecker. 2023. Text compare. Accessed: June 15, 2024.

Stuart J.H. Graham, Alan C. Marco, and Richard Miller. 2015. The uspto patent examination research dataset: A window on the process of patent examination. Technical report.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. LongT5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States. Association for Computational Linguistics.

Jieh-Sheng Lee and Jieh Hsiang. 2020. Patent claim generation by fine-tuning openai gpt-2. *World Patent Information*, 62:101983.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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*, pages 7871–7880, Online. Association for Computational Linguistics.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. Cite arxiv:1907.11692.

Eric Malmi, Yue Dong, Jonathan Mallinson, Aleksandr Chuklin, Jakub Adamek, Daniil Mirylenka, Felix Stahlberg, Sebastian Krause, Shankar Kumar, and Aliaksei Severyn. 2022. Text generation with text-editing models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorial Abstracts*, pages 1–7, Seattle, United States. Association for Computational Linguistics.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5054–5065, Hong Kong, China. Association for Computational Linguistics.

Alan C. Marco, Joshua D. Sarnoff, and Charles deGrazia. 2016. Patent claims and patent scope. USPTO Economic Working Paper 2016-04, USPTO. Available at SSRN: https://ssrn.com/abstract=2844964.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, PA, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. Accessed: 2024-06-12.

Alexander Spangher, Xiang Ren, Jonathan May, and Nanyun Peng. 2022. NewsEdits: A news article revision dataset and a novel document-level reasoning challenge. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 127–157, Seattle, United States. Association for Computational Linguistics.

Mirac Suzgun, Luke Melas-Kyriazi, Suproteem Sarkar, Scott D Kominers, and Stuart Shieber. 2023. The harvard uspto patent dataset: A large-scale, well-structured, and multi-purpose corpus of patent applications. In *Advances in Neural Information Processing Systems*, volume 36, pages 57908–57946. Curran Associates, Inc.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.