Writing Patent Claim with a hybrid AI model : Web Application and Sequence-to-sequence Models

Hannah Moon
Carnegie Mellon University
Pittsburgh, PA 15213
hyunm@andrew.cmu.edu

Yiwei Qin
Carnegie Mellon University
Pittsburgh, PA 15213
yiweiq@andrew.cmu.edu

Ruijia Chen
Carnegie Mellon University
Pittsburgh, PA 15213
ruijiac@andrew.cmu.edu

Ozan Tonguz
Carnegie Mellon University
Pittsburgh, PA 15213
tonguz@andrew.cmu.edu

Abstract

Improving time efficiency has been an important argument in filing patent applications in legal service. A series of Natural Language Processing(NLP) models improved significant time efficiencies. In this paper, we propose a hybrid Web Application and sequence to sequence(Seq2Seq) recurrent neural network(RNN) machine learning to write patent claim. We apply (**describe more about the method). Our research showed that (**Mention about what we achieved throughout the project). We also propose additional extensions to the current model for further iterations in improving (what *****).

Keywords— Deep learning; Neural Network: Natural Language Processing: Patent Claim; Sequence to Sequence(Seq2Seq): Recurrent Neural Networks(RNN)

1 Introduction

In this chapter, we describe the research motivations, objectives, methodology, experiment, analysis and overview of the paper.

1.1 Research Background

Natural Language Processing(NLP) has been widely adopted to speed-up many tasks. Although different NLP methods are being used in many legal services, it is still in infancy in patent legal service. For instance, there are just a few companies that offer products on writing drafts of patent applications whereas other legal services actively implementing innovative NLP methods to optimize services(Dale, R. (2019)).

The main reason is that filing patents is a very complex system that needs to go through rigorous iterative processes by human lawyers to make inventions patentable, therefore, applying a single NLP method is not solving a complex problem if we understand the unique attributes of filing patents and status quo technologies. Due to the slow innovation in patent legal service, human lawyers are writing patent application manually which is inefficient and costly. Therefore, it is necessary to apply new approaches to optimize the process of writing a patent application, especially writing claim part of application should be optimized where lawyers spend majority of time when writing patent applications.

1.2 Research Motivations

We began this research to achieve two main motivations to accomplish in developing a cooperative Web Application and Seq2Seq RNN model. First, we aim to automate writing the patent claim part to increase efficiency of writing patent claim on the patent application. According to United States Patent and Trademark Office(USPTO), the United States is no longer a leading country on generating granted patents.

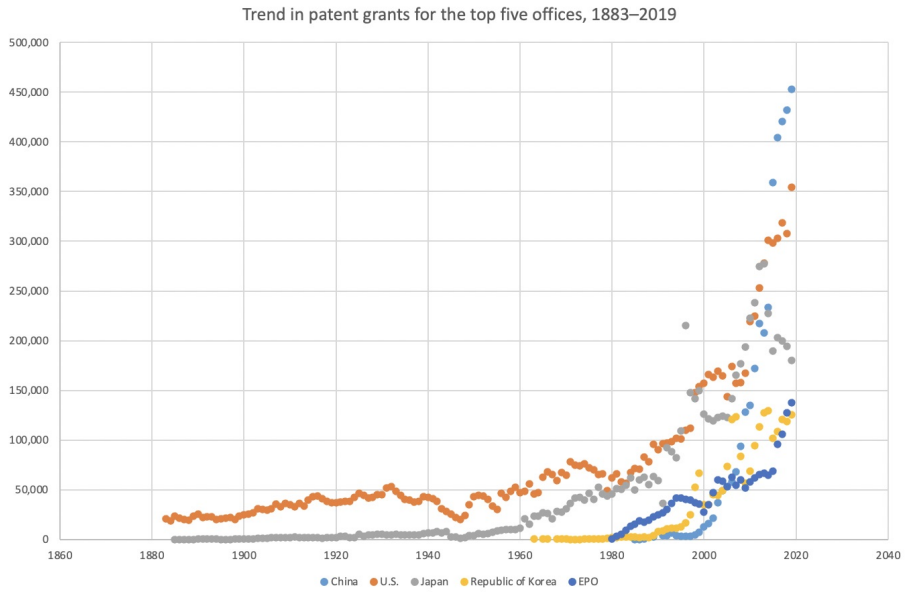

Figure 1: Trend in patent grants for the top five offices, 1883–2019

As protecting intellectual property is the same as securing national wealth(Ash, 2014), the number of granted patents will improve the efficiency of filing patent applications and reduce the amount of time for patent service providers if we automate patent application process which eventually help to generate more granted patents. However, most studies in the patent application automation are focusing on prior art search and data analysis(Aristodemou. 2018), and not many researches discuss writing patent claims automatically. Writing indisputable patent claim is important as it defines boundary of patent, whether it does or does not protected. Accordingly, establishing a hybrid cooperative multi-agent platform to automatically utilize the domain thesaurus and extract the useful knowledge from patent disclosure is an essential.

Another attribution is that we want to find opportunity whether the state of art technology can turn out to assist human patent lawyers to optimize the patent filing process which is written by inventors. An innovative Web-Application empowers inventors to write the quality of claims by themselves. With properly collected information by inventors that are written by similar legal languages, the patent lawyers can spend more time on the place where they are good at by adding more economic value to write patent applications.

1.3 Research Objectives

The main objectives of this research are to develop:

(i) a process to train Seq2Seq RNN model for trading strategies calibration and aggregation. In this process, we show how the training and selection of the generator is designed overall. We aim to optimize or change the combination of models by understanding the clear process of system design for future iteration if necessary.

(i) a Web-Application(WA) that is able to synchronize with Seq2Seq RNN model to feed information to AI model. So once AI model is trained and fully functioning, the synchronized WA is capable to feed tailored information and in which can be embodied in the AI model.

1.4 Research Methodology

We propose a cooperative hybrid platform to generate the first patent claim to speed up the filing patent application process. A solution for this problem is to utilize an optimally coordinated Web Application and Seq2Seq RNN. We focus on the tasks of text generation to map out words from summaries of inventions as an input to generate target sequence of word combinations of patent claim no.1 as an output.

1.5 Structure of Paper

In this paper, we contribute the ground work on proposing the hybrid model approaches to generate first patent claiming and potential implementation of the model. The structure of this paper is as below: In section 2, we explain about related work. In section 3, we describe proposed method of framework of Web-Application, LightSite, and RNN. In section 4, we present experiment of the models. In section 5, we talk about conclusion and future research discussion.

2 Related Work

In this chapter, we describe the body of scientific literature scrutinized by this research. We aim to emphasize the research opportunity gap while in this phase of the research that we need to fill those gap to achieve our research motivation. We reviewed state cuo NLP utilization in legal innovation especially focusing on patent filing automation, then explore to define the ideal models we would employ for experiments. We then move ahead to find the downside of the model we believe it is ideal for this research.

2.1 Status Cuo

With the rise of NLP based on deep learning, many methods have been developed to embody question s and answers based frame work in Web-Application. Especially, Django[5] helps developer to build Web-Application efficiently. Django is a Python web application framework that helps to develop a clean and rapid web-application with less hectic. It is an open source platform, so has been widely adopted to build WebApp. Question and Answer Matching task, the initial method was mainly based on simply asking questions to users to collect sufficient data sets.

As for Language Processing , many authors have addressed the automation of classifying patents [4] or searching prior art[5]. In general, these are necessary steps to write strong patent claims from patent lawyers' perspectives.

However, when we find distinctive approaches in Natural Language Processing(NLP), we are able to find more lucrative research activities. Based on the literature review, we can outline three distinct approaches with utilizing NLP in Intellectual Property: Data Search, Data Analysis, Data-Driven Product Development, and patent retrieval based on Data.

M.Lupu[6] pointed out that among patent-related applications, modern neural networks are mainly applied to machine translation, and that there is a wide range of opportunities for other tasks such as patent analysis, patent evaluation, and patent classification. He also expected that the surprising success of deep learning would one day be tested with patent data. In computer science, natural language processing (NLP) converts text into structured data, and natural language generation (NLG) converts structured data back to text. In recent years, transfer learning based on Transformer models such as Seq2Seq RNN and Transformer have significantly outperformed various tasks after using pre-trained language models for large corpus.

We see it as an opportunity that the successful application of NLP method were generated claims. In this work, we promote to apply the latest NLP Seq2Seq model to generate patent claim.

2.2 Natural Language Processing(NLP)

Natural Language Processing(NLP) can be divided into two main tasks: Natural Language Understanding(NLU), which turns text into structured data, and Natural Language Generation(NLG), which turns structured data back to text. In recent past, Seq2Seq model, a deep-learning based model that maps an input sequence into another output sequence, which is a typical combination of NLU and NLG process, has been successful in many problems such as machine translation[7], and text summarization[8].

Since filling patents is a very complex system, we narrow down our problem to generate the first independent claim with description of invention files provided by inventors. This task is very similar to text summarization which combines NLU to read a long document and NLG to generate target text.

In the framework of sequence-to-sequence models, the attentional Recurrent Neural Netword (RNN) encoder-decoder model proposed by [4] is relevant to our task. Although attentional RNN is designed for machine translation, which is a very different problem from our task; based on the similarities, we decided to use this model as our first attempt to solve our task. Besides, transfer learning based on Transformer models[6], such as GPT, BERT, and GPT-2, has a great success in many tasks after using pre-trained language models on large-scale corpora. We fine tuned a GPT-2 model on our patent dataset to obtain a language model specified for patent claims.

3 Framework

3.1 Web application

We built a web application via Django framework to collect invention disclosure from inventors with the goal of collecting data via WebApp. The accumulated data will be used as a source of input data for NLP model in the later phase of the research.

Once inventors enter into web-application interface, a list of questions were presented to inventors. The visitors of the website were guided to fill out the blanked answer parts. The back-end of the web application integrates the answers as input for generating claims. For now, the background integration operation we are doing is simply to combine the answers of the questions.

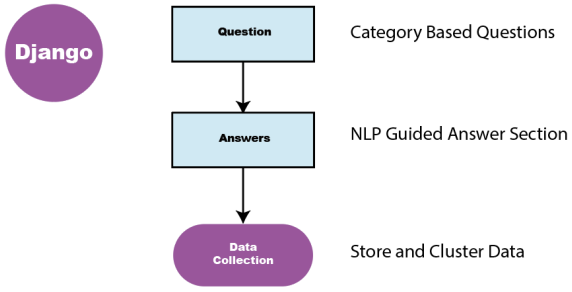

Figure 2: Django Framework Summary

3.2 RNN encoder-decoder

Here, we describe briefly the RNN encoder-decoder model with word attention, proposed by [6], which we used to generate claims. The structure is showed in Figure 3. We denote the input or source sequence as $X = (x_1, x_2, ..., x_N)$ and denote the output or target sequence as $Y = (y_1, y_2, ..., y_T)$. In our task, X is the invention description and Y is the claim. We follow the conditional chain rule of probability to model:

$$p(Y|X) = \prod_{t=1}^{T} p(y_t|X, \{y_1, y_2, ..., y_{t-1}\}), \tag{1}$$

The encoder reads X and represent it as a context vector c. In our case, we use RNN such that

$$h_t = f(x_t, h_{t-1}) \tag{2}$$

and

$$c = q(h_1, h_2, ..., h_T), \tag{3}$$

where $h_t \in \mathcal{R}^n$ is a hidden state at time t, c is a vector generated from the sequence of the hidden states and f,q are some nonlinear functions. We used bidirectional LSTM as f. In a model without attention $c = h_T$.

The decoder is often trained to predict the next word y_t given the context vector c and all the previously predicted words $(y_1, y_2, ..., y_{t-1})$. In other words, the decoder defines a probability over Y:

$$p(Y|X) = \prod_{t=1}^{T} p(y_t|c_t, \{y_1, y_2, ..., y_{t-1}\}), \tag{4}$$

With RNN, each conditional probability is modeled as

$$p(y_t|c_t, y_{<t}) = g(c_t, y_{t-1}, s_t), \tag{5}$$

where g is a nonlinear function that outputs the probability of y_t, and s_t is the hidden state of the RNN.

In the attention model, the context vector c_t is distinct for each target word y_t and it depends on the whole hidden state sequence $(h_1, h_2, ..., h_T)$ to which an encoder maps the input sentence. c_t is calculated by a weighted sum of the hidden states:

$$c_t = \sum_{j=1}^{T} \alpha_{tj} h_j. \tag{6}$$

The weight is computed by

$$\alpha_{tj} = \frac{exp(e_{tj})}{\sum_{k=1}^{T} exp(e_{tk})}, \tag{7}$$

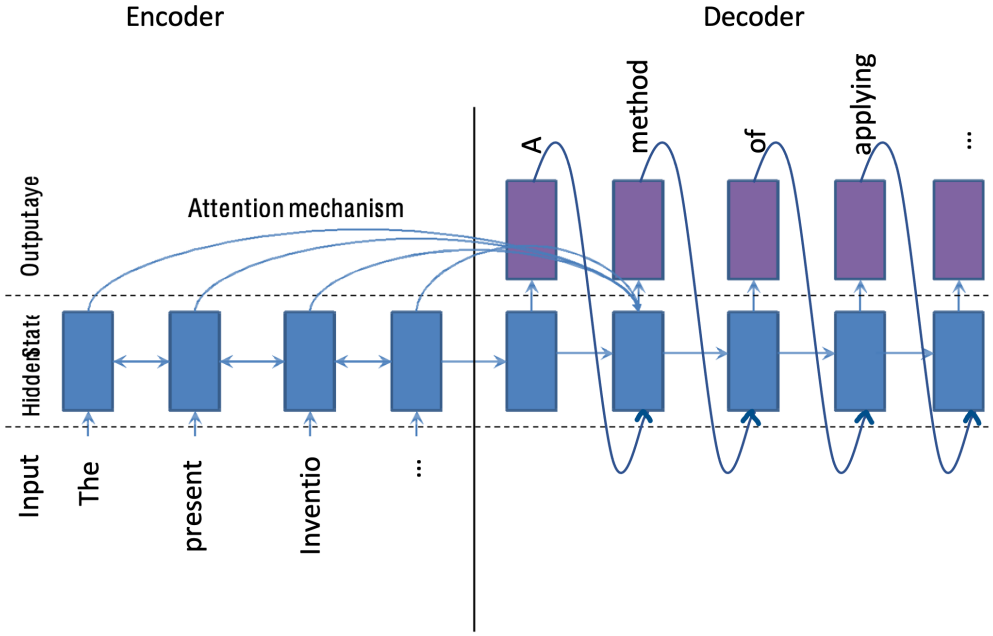

Figure 3: RNN Encoder-Decoder

where $e_{tj} = a(s_{t-1}, h_j)$ scores how well the j-th inputs and the i-th output match. The score is based on the previous RNN hidden state s_{t-1} and the j-th input. We use dot product as the score function in our experiment.

During training, we optimize over the model parameters θ the sequence cross-entropy loss:

$$l(\theta) = -\sum_{t=1}^{T} logp(y_t|X, y_{<t}, \theta), \tag{8}$$

thus maximizing the log-likelihood of the training data.

4 Experiments

We set three goals in our experiments. The first goal is to understand claims by observing the claim text. The second goal is to execute expiatory data analysis to define the frequently used language to generate relate questions for WebApp. Final goal is to compare the generated patent claims based on different combinations of NLP methods with random data.

4.1 Web application

The first experiment is that we read a lot of claims by ourselves, trying to find out their similarities (structure, content, distribution, etc.) and come up with some questions by ourselves. We hope that the answers from the clients can be directly used as claims or input of model in machine learning part which can make the prediction results of the model more accurate.

4.2 Exploratory Data Analysis

The second experiment is using LightSite software to define what is the most likely word, word combinations, and sequences. In this experiment, we used Bayes' rule.

$$P(Words|signal) = P(signal|Words)P(Words) \tag{9}$$

We decompose first claim data into LightSite to classify the frequently us from claim no.1. We designed this step of experiment to ask relevant questions for the web-app development.

(1) What is in this corpus of claims?

Unigram Analysis		Bi-Gram		Tri-Gram	
comprising	238	comprising	754	at_least_one	144
wherein	198	wherein	865	a_plurality_of	126
configured	97	configured	346	BOL_a_method	120
device	81	device	160	and_wherein_the	104
including	65	with_the	151	of_the_first	94
system	65	method	145	and_a_second	93
method	64	plurality	509	one_of_the	87
based	62	including	127	one_or_more	81
plurality	58	a_method	123	the_plurality_of	70

Figure 4: n-Gram Analysis

(2) What is the relationship between corpora of documents?

For question number number one, we experimented with extraction of features from the text data, and use of these features in classification algorithms to learn patterns that associate each claim. Such patterns might be used by patent lawyer to automatically classify and route new claims. We compute the weight of each words as below:

$$Freq(w) = (1Freq) + Freqtf(w) \tag{10}$$

As for the second question, the term frequency can be calculated in both corpora as below:

$$Corelations = \frac{tF_C(w)}{tF_C(w) + tF_S(w)} \tag{11}$$

We prepared 556 rows of data set. We first decided to extract three different features, Unigrams, Bigrams, and Trigrams because we wanted to see how number of word combinations varies helps to understand the content of the claims. Then, before we begin the data extraction, we eliminated "punctuation" as it is unnecessary for the sake of data extraction efficiency.

From the exploratory data analysis, we found that unigram and bigram for both total and target features are quite similar with minor differences. As for the unigram, the most of top 10 words were begin with different pronouns due to the nature of English. The articles "the" or "a" were used frequently, so we eliminated not so important words from the list. We then get top 10 words across all claims. In all total complaints, the top 10 words are like Figure No.4.

After defined key words, we generate candidate of questions.

- What is the invention?
- What method / device / tool / system is the invention comprising of?
- What does the invention would configure?
- What does the invention include?
- Can you describe plurality of the invention?

4.3 Machine Learning

4.3.1 Datasets

Since we would like to find if we could use natural language processing technology to generate a claim from the summary part of the client's documents. We hope to find a ready-made dataset to support our experiment. Unfortunately, after a lot of resource seeking and filtering, we didn't find a dataset that exactly fit our needs. But, We found that Google has a dataset named Patents Public Datasets on BigQuery, with a collection of publicly accessible. Thanks to this user-friendly platform, we could easily get the id of patents for decades. And then, We did data mining via crawler technology implemented by WebDriver framework to crawl the entire published documentations on the United States Patent and Trademark Office(USPTO) website. and then extract the parts we need. Finally, we cleaned the data utilizing a library named beautiful soup which is a famous JSON file processing library to get our own dataset.

Datasets	# Docs(train/val/test)	avg. description words	avg. claim words
IP-set	38,018/389/167	1057	224

Table 1: Dataset Statistics

% uni-gram intersection in gold claim	% bi-gram intersection in gold claim	% tri-gram intersection in gold claim
73.8	55.4	43.9

Table 2: n-gram intersection

Our training dataset contains 38,018 pairs of patent invention summary and patent claims of the U.S. patents. All the claims are the first independent claims. Table 1,2 shows the statistics of our dataset.

4.3.2 RNN Encoder-Decoder

In this experiment, we built an attentional RNN Encoder-Decoder proposed by [4]. We used the invention summary as our input description file. Here is an example of a description file, gold standart claim and our generated claim:

description file: *The present invention relates to a lock including a hollow barrel that is secured around an IV hub . Seated within an open upper end of the barrel are a pair of concave , hinged gates that move between a horizontal , closed position and a substantially vertical , open position . Below the gates are a pair of spring-biased , arcuate flaps that normally prevent the gates from lowering . When a key is inserted into arcuate slots on the gates and rotated , the flaps are forced outwardly toward the barrel periphery , allowing the gates to open with a downward force . When the key is rotated in an opposite direction , the flaps automatically pivot toward the center of the barrel , lift the gates to the closed position and prevent them from reopening when the key is removed . It is therefore an object of the present invention to provide a lock that prevents unauthorized substances from being injected into an IV site . It is therefore another object of the present invention to provide a lock for an IV hub that can only be opened with a corresponding key . Other objects , features , and advantages of the present invention will become readily apparent from the following detailed description of the preferred embodiment when considered with the attached drawings and the appended claims .*

gold standard claim: *1 . An IV lock comprising : an IV hub having an injection port thereon ; a barrel enclosing said hub , said barrel having an upper end and a lower end with the upper end proximal said injection port ; a locking means within the upper end of said barrel for restricting access to said injection port , wherein said locking means includes a pair of hinged gates that are movable between an open position and a closed position , said hinged gates blocking said injection port when in the closed position , a pair of spring-biased flaps engaging said gates to maintain said gates in the closed position , and a means for displacing said spring-biased flaps out of engagement with said gates to allow said gates to open upon application of a force thereto ; a key for disabling said locking means .*

generated claim: *1 . A lock , comprising : a hollow barrel that is secured around an IV hub ; and a spindle within an open upper end of the barrel and a pair of concave , hinged gates that move between a horizontal , closed position and magnetic-flux vertical , open position ; wherein when the key is rotated in an opposite direction , the flaps automatically pivot toward the center of the barrel , lift the gates to the closed position and prevent self-interference securement when the key is removed .*

From the above example, we found that our model can generate text in claim format and is related to the topic of the description file. We cherry picked a good example that makes some sense, but there are many meaningless samples in our experiment. Although the result is not good, this experiment provides us valuable experience to guide our next-step model design. By looking at the results, we found the following issues:

1. In our experiment, we only used the summary of a patent as the description file. However, some invention summaries are really short, e.g. in some cases, the summary is only one sentence. Those short summaries cannot provide enough information to generate the claims. The possible solution is changing the input. We have come up with three possible inputs: a) Include more information in our input. Up till now, we only use the invention summary of a patent document and ignore a lot of information included in the patent document, so it might be helpful to add invention abstract, invention background or other parts of a patent document to our input. b) Combine with web application part and use the answers of the web application questions as the input. c) Combine extractive and abstractive models similar to what have proved to be effective in text summarization task. Firstly uses an extractive model to extract the most important sentences from the whole patent file and then uses a sequence to sequence architecture to generate claims with the extracted sentences as input.

2. We noticed that when the input summary is long, the model will generate meaningless claims. The possible reason is that the model cannot capture the right information from the description file when it is too long. The naive model we use, RNN seq2seq model with word attention, is originally proposed for translation, whose source and the target is only one sentence. However, in our dataset, the input is hundreds of words and the output is typically over one hundred words. Besides, in translation, there is a strong one-to-one word-level alignment between source and target and it is less obvious in our task. Thus, the word-level encoder and attention mechanism might not be appropriate in our task. In the long source document, in addition to identifying the keywords, it is important to identify the key sentences from which the claims can be drawn[5]. In our future experiment, we will use both word-level encoder and sentence-level encoder to represent the description file and use the attention mechanism operating at both levels simultaneously.

3. We noticed that our generated claims suffer from repeated words or sentences, syntactical and semantical errors. The possible solution is to replace the RNN-based decoder with a more powerful transformer-based decoder which has proved to outperform RNN, e.g. GPT-2 that we have experimented on in the next section.

4.3.3 Fine-Tuning OpenAI GPT-2

In this experiment, we fine tuned an OpenAI GPT-2 pretrained model to generate patent claims. OpenAI built four GPT-2 models whose sizes are 117M, 345M, 762M and 1.5B, in terms of the number of parameters in the neural network. In general, larger model will generate better results. However, due to the limitation of computer resource, we used the model of 117M, which has 12-layer, 768-hidden, 12-heads, and found that it is enough to generate impressive results. After training three epochs on the 38,018 claims of our dataset, we generated the following text:

1. A semiconductor device comprising: a first channel formed in a first direction; a second channel formed in a second direction crossing the first direction in a @@@ field-of-view (FOV) scanning direction; a driving unit coupled to the first channel, and coupled to the second channel via the first channel and the second channel; a gate driving unit coupled to the driving unit via the first channel, and coupled to the @@@ second channel via the second channel, wherein the second channel, the first channel, the driving unit, and the gate driving unit are formed as a single unit; and an under layer comprising a first oxide layer, a second oxide layer and a third oxide layer, wherein the @@@ underlayer comprises a second oxide layer, the third oxide layer is disposed between the first oxide layer and the second oxide layer, and the first oxide layer and the third oxide layer are disposed adjacent to each other in the FOV scanning direction.

The special token "@@@" means a new line. We adopt the same segmentation token to decompose the lengthy patent claims into multiple claim spans as [7]. We observed from the above generated example that our model can generate coherent claims. Although this GPT-2 model, which only generate claims randomly, is not conditioned on the description file to generate something related to a specified patent, and this experiment is not exactly what we need to solve our task, it is good to see our language model can generate claims of acceptable quality having some practical meaning and having no obvious syntactical or semantical error. We believe this language model will be useful to replace a RNN decoder of the sequence to sequence model in our future experiment.

5 Conclusion

A hybrid Web-App and Seq2Seq for writing first patent claims was presented in this paper. However, claims are not written in the same style of English as they appear in news articles or on the Web. Therefore, different combinations of NLP should be considered to generate claims before developing Web-Applications to aid human lawyers to write patent claims accurately while improving time efficiency.

6 Future Work

Our work also opens new avenues to the direction of future research opportunities as follow:

• Experiment 1: a new model to automate with patent application drafting, in which it presents the likelihood to human patent lawyers' written claims based on given the amount of raw data used. We need to outline the parameter to measure the accuracy based on the sample results.

• Experiment 2: Human-Computer Interactive Questions and Answer Web-Application - Directly translate inventors' scientific language into legal languages to generate claims.

• Experiment 3: Different Architecture - Due to the issues emphasized in section 4, use the hierarchical encoder and attention structure, replace RNN-based decoder with transformer-based decoder, and use different inputs.

References

[1] Dale, R. (2019). Law and word order: NLP in legal tech. Natural Language Engineering, 25(1), 211-217. doi:http://dx.doi.org.proxy.library.cmu.edu/10.1017/S1351324918000475

[2] Ash, Reggie. "Protecting Intellectual Property and the Nation's Economic Security." Landslide (Chicago, Ill.) 6.5 (2014): 20–. Print.

[3] Aristodemou, Tietze. "The State-of-the-Art on Intellectual Property Analytics (IPA): A Literature Review on Artificial Intelligence, Machine Learning and Deep Learning Methods for Analysing Intellectual Property (IP) Data." World patent information 55 (2018): 37–51. Web.

[4] Smith, H. (2002). Automation of patent classification. World Patent Information, 24(4), 269-271.

[5] Django. Design philosophies, 2015. URL https://docs.djangoproject.com/ en/1.10/misc/design-philosophies/. Accessed: 2017-02-20.

[6] Chin, A. (2008). Search for tomorrow: some side effects of patent office automation. NCL Rev., 87, 1617.

[7] M.Lupu, Information retrieval, machine learning, and Natural Language Processing for intellectual property information, World Pat. Inf. 49 (2017) A1–A3. doi:10.1016/j.wpi.2017.06.002.

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473.

[9] Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond

[10] A.Vaswani, N.Shazeer, N.Parmar, J.Uszkoreit, L.Jones, A.N.Gomez, L.Kaiser, I.Polosukhin, Attention Is All You Need, (2017).

[11] Patent Claim Generation by Fine-Tuning OpenAI GPT-2