Court Opinion Generation from Case Fact Description with Legal Basis

Quanzhi Li, Qiong Zhang
Alibaba Group
Bellevue, WA 98004, USA
{quanzhi.li, qz.zhang}@alibaba-inc.com

Abstract

In this study, we proposed an approach to automatically generating court view from the fact description of a legal case. This is a text-to-text natural language generation problem, and it can help the automatic legal document generation. Due to the specialty of the legal domain, our model exploits the charge and law article information in the generation process, instead of utilizing just the fact description text. The BERT model is used as the encoder and a Transformer architecture is used as decoder. To smoothly integrate these two parts together, we employ two separate optimizers for the two components during the training process. The experiments on two data sets of Chinese legal cases show that our approach outperforms other methods.

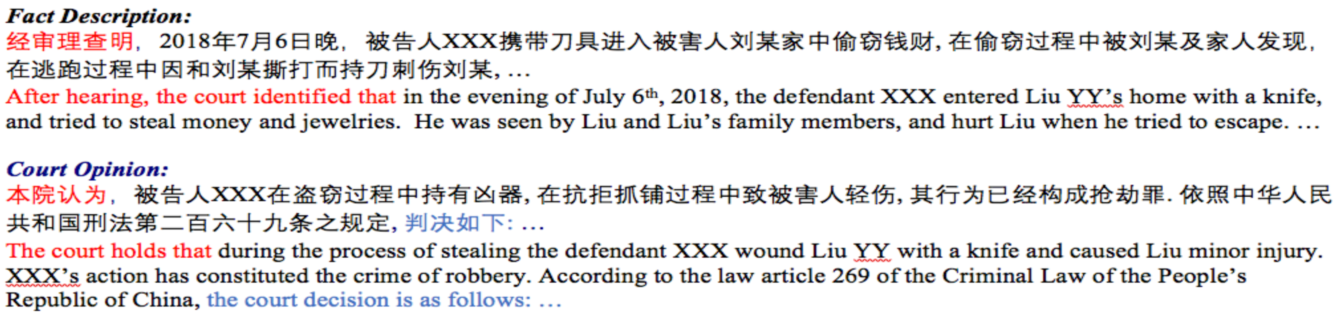
Introduction

Previous studies in legal domain mainly focus on classifying legal documents, retrieving relevant cases, and predicting charge labels and law articles given the fact description of a case. An area that has not been explored much is the automatic generation of legal text, especially the text that have latent legal logic or knowledge, such as the court opinion section of a written judgment.

One of the most common legal document types is the document of written judgment. Almost for every case that is processed by the court or judges, there will be such a document. In the Chinese legal system, it is called "判决书" or "裁判文书". In the judgment document, there is always a section called "court view" or "court opinion" ("本院认为" in Chinese), which explains the charge decision by the judges and usually also cites the applied law articles. This section is the core part of a legal judgment document. Another important section is the "fact description" ("案件事实") part, which describes the facts of the case. Judges write the court view part based on the case facts, relevant law articles and charges. And following the court view part, there is a section called "court decision" or "judgment" ("判决结果"), which is the court's final decision based on the rationales described in the court view section. An example of a Chinese case judgment document is shown in Figure 1. This figure shows two parts of the judgment document, the fact description and the court opinion parts. The words in red indicates that it is the start of the corresponding part (fact description or court view). And the words in blue indicate the start of the final decision part.

This study focuses on generating the court view part, excluding the court final decision part. Automatic court opinion generation has many benefits and real applications. For example, it can reduce the workload of judges on writing court opinions, especially for simple cases but in large amount. It also benefits the individuals seeking legal advices. Nowadays, more and more people seek legal advice online. One common scenario is that the individual writes a paragraph of text that describes the facts, and wants to know the opinion of judges or legal professionals on that. Automatic court opinion generation is a good tool for this type of scenarios, especially for types of legal cases that are simple but with a large volume. Our algorithm is being deployed in several Chinese intelligent court systems and some legal consulting service platforms. These legal consulting service systems provide free online services for individuals, especially for the ones in the remote areas in China, where it is hard for low-income people to access in-person legal service, due to its expensiveness and the lack of legal professionals.

Court view generation is a text-to-text generation problem, but compared to other text generation problems, such as text summarization, it is more challenging, since it needs to deduce the latent information from the facts, and generate the opinion text that is both law and charge discriminative. In order to generate such type of court view, we proposed a court opinion generation model that has the following features: 1. It is based on the Transformer (Vaswani et al., 2017) architecture, but we use the BERT model (Devlin et al., 2019) as the fact description and law article encoder, to



Fact Description:

经审理查明，2018年7月6日晚，被告人XXX携带刀具进入被害人刘某家中偷窃钱财，在偷窃过程中被刘某及家人发现，在逃跑过程中因和刘某撕打而持刀刺伤刘某，…

After hearing, the court identified that in the evening of July 6th, 2018, the defendant XXX entered Liu YY's home with a knife, and tried to steal money and jewelries. He was seen by Liu and Liu's family members, and hurt Liu when he tried to escape. …

Court Opinion:

本院认为，被告人XXX在盗窃过程中持有凶器，在抗拒抓铺过程中致被害人轻伤，其行为已经构成抢劫罪. 依照中华人民共和国刑法第二百六十九条之规定，判决如下：…

The court holds that during the process of stealing the defendant XXX wound Liu YY with a knife and caused Liu minor injury. XXX's action has constituted the crime of robbery. According to the law article 269 of the Criminal Law of the People's Republic of China, the court decision is as follows: …

Figure 1: A legal case example, showing the fact description and court view sections.

better exploit the grammar and semantic information of tokens learned by the pre-trained model. 2. Since the BERT encoder is pre-trained and the decoder is not, in order to smoothly integrate these two parts together for better generation performance, we employ two separate optimizers for these two components during the training process. 3. To make the court view more law and charge discriminative, we utilize the charge and law article information in the generation process; they are encoded and attended by the Transformer decoders. The above three points are also the main contributions of this study.

Automatic law article prediction has been studied by several studies, and there are already many studies on charge label prediction, so these two prediction tasks are not the focus of this study. In this study, we extract the charge label and law articles from the case documents. Jointly learning the court view generation and prediction of charge label and law articles will be one of our future studies.

Related Studies

Previous studies on legal assistant system mainly focus on the fields of legal case retrieval (Chen et al. 2013; Raghav et al. 2016), case classification (Nallapati and Manning, 2008; Wang et al. 2018), legal case data set analysis (Locke and Zuccon, 2018; Du et al. 2019), charge label prediction and law article prediction (Liu and Liao, 2005; Liu and Hsieh, 2006; Liu et al. 2015; Luo et al. 2017; Yang et al. 2019b; Zhou et al. 2019). They use either the traditional machine learning models or neural networks. Ye et al. (2018) use LSTM based encoder-decoder model with attention (Sutskever et al. 2014; Vaswani et al. 2017) to generate court view from fact description.

Sequence to sequence (seq2seq) learning has been used in a variety of language generation applications. It has attracted much attention in recent years due to the advance of deep learning. Our model also belongs to this widely used seq2seq paradigm (Sutskever et al. 2014). Rush et al. (2015) and Nallapati et al. (2016) were among the first to use the neural encoder decoder structure in text summarization. See et al. (2017) enhance this model with a pointer generator network which allows it to copy words from the source text. Paulus et al. (2018) present a deep reinforced model for abstractive summarization which handles the coverage problem with an intra-attention mechanism. Celikyilmaz et al. (2018) propose an abstractive system where multiple agents represent the document together with a hierarchical attention mechanism for decoding. Narayan et al. (2018) propose an abstractive model suitable for single sentence summaries.

Pre-training has been widely used in natural language processing (NLP) tasks to learn better language representation, and several new pre-trained models have been published recently, such as BERT (Devlin et al. 2019), XLNet (Yang et al. 2019a), RoBERTa (Liu et al. 2019), ALBERT (Lan et al. 2019), ELMO (Peters, et al. 2018), etc. The pre-training on large amount of unlabeled data and fine-tuning with small scale labeled data are helpful for many tasks, and it is also used in the encoder part of our model in this work. Devlin et al. (2019) proposed BERT based on masked language modeling and next sentence prediction, and achieved state-of-the-art results on multiple NLP tasks. There are also some works on pre-training the encoder-decoder model for language generation (Rothe et al., 2019; Edunov et al., 2019; Liu and Lapata, 2019; Song et al., 2019). The main difference between our model and others are that our model uses pre-trained BERT model in the encoder side and uses a non-pre-trained Transformer on the decoder side, and we fine-tune the encoder and train the decoder using two separate optimizers. More importantly, we utilize the charge and law article information in the decoding process, in order to generate judge opinions that have more legal basis.

Opinion Generation Model

In this section, we first describe the high-level encoder-decoder structure, and then present the fact description encoder, law article encoding, and the Transformer decoder in details. Figure 2 shows the high-level model structure. The fact description is encoded by a pre-trained BERT model, shown on the left side of the diagram. The charge label embedding is learned during the training process. Each case may have multiple law articles applied, and their embeddings are aggerated together to one vector representing all the articles for this case. The right side is the decoder part, which is based on the Transformer architecture, consisting of 6 layers of decoders. The fact embedding, charge embedding and the aggregated law article embedding are fed to each of the decoder layers, and used by the multi-head attention layer described later. The charge embedding and law article embedding are also used at the final softmax layer of the Transformer decoder, in order to add more context to help the decoder choose the correct token.

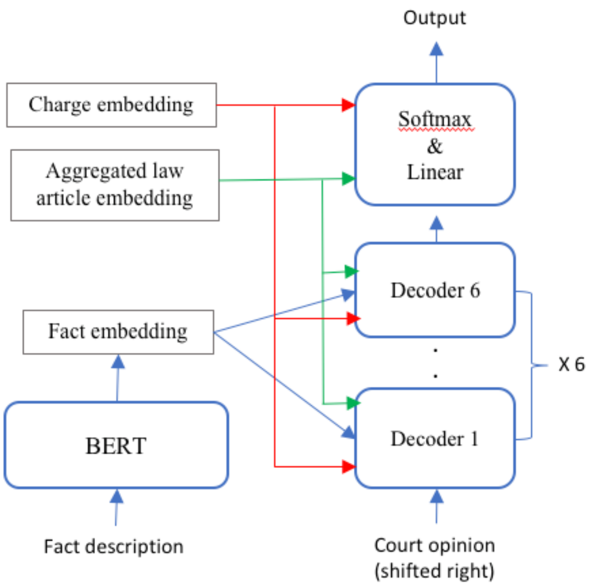


Figure 2: The high-level encoder-decoder structure.

Fact Description Encoder and Law Article Encoder

Fact Description Encoding.
We use BERT to encode fact description and law articles. BERT has been used to fine-tune various NLP tasks, but its application to text generation is not straightforward, since it is trained to predict single word and next sentence, not generating text sequence. This is why our model uses BERT to encode the fact description, but utilizes Transformer on the decoding side to generate text sequence. The BERT-Chinese model is fine-tuned by the fact descriptions from the training data, and the output from the last layer is fed to the decoder side.

Law Article Encoding.
In the Chinese legal system, each law article has a description defining the content of that article and in what situations that article should be applied. When judges express their view on a case, their opinions are based on the facts and the applied law articles, and the languages they use are usually affected by the definition of the related law articles. Sometimes some sentences or phrases in the court view part are the same as the sentences or phrases in the articles. Therefore, we hypothesize that the textual content of law articles will help the court view generation.

For a law article, we also use the pre-trained BERT-Chinese model to generate its embedding by fine-tuning the model. Since a case may involve multiple law articles, we aggregate these law article embeddings, and generate a final representation and feed it into the Transformer decoder. For each article, we take the [CLS] token vector from the last layer to represent the article. And then for all the articles of this case, we take the average of their embeddings. The final embedding is fed into the decoder side for multi-head attention.

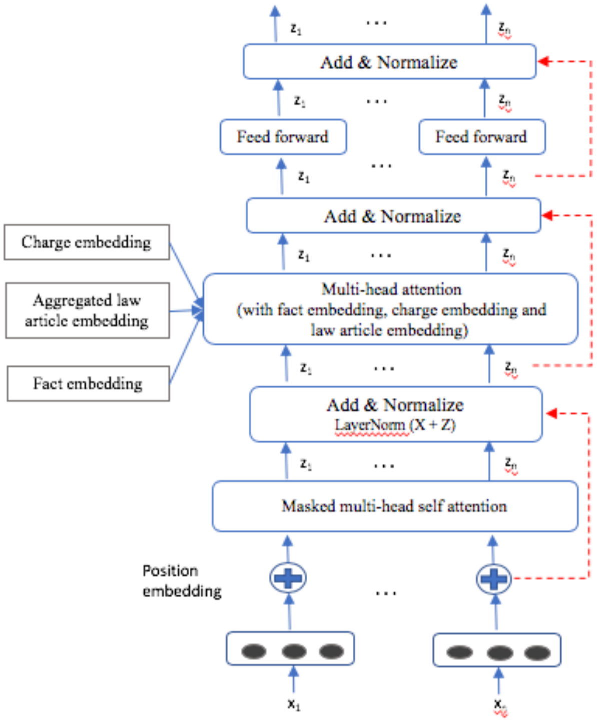

z₁
...
zₙ
Add & Normalize
Feed forward
Feed forward
Add & Normalize
Charge embedding
Aggregated law article embedding
Fact embedding
Multi-head attention (with fact embedding, charge embedding and law article embedding)
Add & Normalize
LayerNorm (X + Z)
Masked multi-head self attention
Position embedding
x₁
xₙ

Figure 3: Detailed structure of a decoder layer. This diagram shows the first decoder layer.

Transformer Decoder

Figure 2 shows that there is a stack of six decoders in the decoding side. Figure 3 presents the detailed structure of a decoder, using the first decoder layer as an example. There are two multi-head attention layers in this decoder, one is a masked multi-head self-attention, and the other one is the decoder attended on the three types of contexts, i.e. the fact description embedding, charge label embedding and law article embedding. After each attention layer and the feedforward layer, there is a Add&Normalize layer.

Multi-head Attention.
In the multi-head attention structure, attention is computed not once but multiple times, in parallel and independently.

The outputs are concatenated and linearly transformed, as shown in Figure 4. In this figure, *m* is the number of heads. Figure 5 shows how one head of the scaled do-product attention is computed, which can be expressed as follow:

$$Attention\ (Q, K, V) = softmax\ (\frac{QK^T}{\sqrt{d_k}})\ V \qquad (1)$$

where *Q* is a matrix that contains the query (vector representation of one word in the sequence), *K* are all the keys (vector representations of all the words in the sequence), and *V* are the values. For the first multi-head attention layer in each decoder, *V* consists of the same word sequence as *Q*. However, for the second attention module in each decoder, it considers the decoder sequence, the fact embedding, charge embedding and law article embedding, and therefore, here *V* is different from the sequence represented by *Q*.

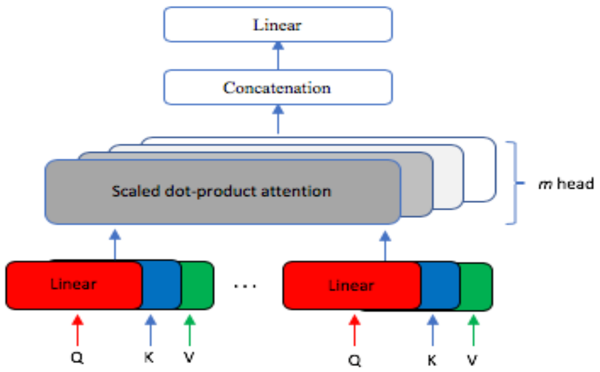


Figure 4: Multi-head attention layer

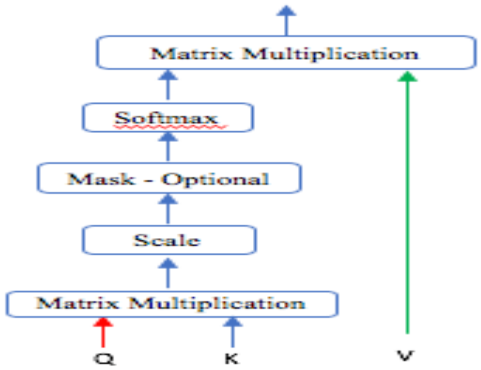


Figure 5: Scaled dot-product attention

Model Training and Inference

In our model, the encoder is based on a pre-trained BERT model, and the decoder component is not pre-trained. It is obvious that there is a mismatch between these two components, because one is pre-trained and the other one needs to be completely trained. This may make the training process unstable, e.g. one component is underfitted and the other one is overfitted. One way to handle this is to use two different optimizers for the two components. In our model implementation, two Adam optimizers are used. Each has its own learning rates and warmup steps. These values will be set so that the pre-trained BERT model should be fine-tuned with a smaller learning rate and decay, when the decoder becomes stable. This is to make sure that the BERT model to be fine-tuned with more accurate gradients. The learning rate update is illustrated by the following equation:

$$learnRate = learnRate * min\ (s^{-0.5},\ s * w^{-1.5}) \qquad (2)$$

where *s* is the step and *w* is the warmup value.

For inference, we use beam search, whose size is set to 4, to find the best sequence. The generated word sequences will be ranked and the one with the largest value will be chosen.

Table 1: Statistics of the two data sets

Data set attribute	Data set 1	Data set 2
Training data	521,500	153,706
Validation data	25,000	9,152
Test data	30,000	9,123
Average number of tokens in fact description part	248.2	219.9
Average number of tokens in court opinion part	34.4	30.6

Experiments and Results

Data Set Preparation and Evaluation Metrics

Data Sets.

Our experiments of comparing different generation approaches were conducted on two data sets. Data set 1 was created by ourselves, and data set 2 was from (Ye et al., 2018). The data in both data sets are originally from the legal documents in China Judgements Online (http://wenshu.court.gov.cn). There is no duplicate between these two data sets. Data set 2 contains mainly criminal cases, while data set 1 contains both criminal and civil cases, with criminal cases consisting of three-fifths of them. The basic statistics of the two data sets are presented in Table 1. We created data set 1 is because that data set 2 contains only criminal cases, the court opinion part is usually short and some of them do not have law articles. The total number of unique law articles in these two data sets are 308, and it is 78 for the unique charges. On average, each case has about two law articles applied. In this study, we only focus on the cases with one charge; the complicated cases are left for future study.

Data Preparation.

Similar to previous studies on Chinese legal cases, we extract the fact descriptions, court opinions, law articles and charge labels using regular expressions. These cases are written by judges, and the format are pretty standard. The paragraph started with “经审理查明” (“the court identified that”) is extracted as the fact description. The part between “本院认为” (“the court holds that”) and the decision “判决如下” (“the court decision is as follows”) are extracted as the rationales/opinions/views. Nearly all the samples in the two data sets match this pattern. In order to meet the text length limitation of pre-trained BERT, a threshold of 510 tokens is set up for both fact description and court opinion parts. Facts and opinions longer than that will be stripped, and they will be studied in the future. Charges and law articles are also easy to be extracted using regular expressions.

Special Tokens.

We use “<DATE>” to replace dates in the data sets. In the Chinese legal system, numbers and entity names may affect the applied laws, charges and the final decision. For example, for a drug trafficking or bribery case, the amount of drug and money will definitely affect the charge, applied law articles and the number of years in prison. To handle this, we use a data-driven approach – we extracted all the numbers from the two data sets, and based on the analysis of these numbers, we split them into 20 buckets based on their magnitudes. These 20 buckets are represented by 20 symbols. Types of entities may also affect a case in terms of the charge, applied law articles and the final judgment. For example, in a bribery case, a government employee and a private company employee would get different charges. To handle this case, we converted the entities of private companies, state-owned companies and government agents to three different symbols. Government agents are easy to identify by using regular expression and rules. We use a lookup service, which provides lookup of most state-owned companies, to check if a company is state-owned. Individual names are converted to special symbol.

Law Article Data Set.

As mentioned before, we need the definition of law articles to generate their embeddings. We processed 308 law articles. Their descriptions/definitions are collected from the website of the Chinese Highest Court.

Evaluation Metrics.

To evaluate the effectiveness of our approach, we use a quantitative comparison between the true court opinion, the one extracted from the real case, and the automatically generated one. This approach has been used by many previous studies in text summarization tasks. We use both BLEU-4 score (Papineni et al. 2002) and three ROUGE scores (Lin and Hovy, 2003) as the evaluation metrics.

ROUGE works by comparing an automatically produced text against the reference text. It basically measures the overlap of N-grams between the system and reference texts. However, the original ROUGE measure does not tell you much as a metric. To get a good quantitative value, in the context of ROUGE, we compute precision and recall using the overlap, and then report the F1-measure of ROUGE. In this study, we use ROUGE-1, which is based on unigram overlap, ROUGE-2, which is based on bigram overlap, and also ROUGE-L, which measures the longest common subsequence of reference and generated texts, to compute the F1 scores.

Compared Methods and Experiment Settings

Compared Methods.

To evaluate our proposed approach, we compared it to the following approaches, including the basic approaches and the state-of-the-art ones.

- **Random & Random$_{\text{+charge\&law}}$**: this method is to randomly select court opinions from the training set. The Random$_{\text{+charge\&law}}$ version will select court view from the training cases that have the same charge label and law articles as the test case. We want to see how much improvement we can get by limiting the case pool to the ones with same charge and law articles. Random method is to give the low bound performance of the court opinion generation task.
- **BM25 & BM25$_{\text{+charge\&law}}$:** this method uses BM25 score (Robertson and Zaragoza, 2009) to retrieve the case whose fact description has the highest BM25 score, and use its court view as the result. The rationale behind this method is that similar fact descriptions may have the similar court views. Similar to The Random$_{\text{+charge\&law,}}$ The BM25$_{\text{+charge\&law}}$ will retrieve the court view from the training cases that have the same charge label and law articles as the test case.
- **Seq2seq-LSTM-attention**: this approach is based on the seq2seq model with attention, and both the encoder and decoder use a bidirectional-LSTM model (Sutskever et al., 2014; Vaswani et al., 2017; Ye et al., 2018). Attention mechanism can catch the important input information for the current output sequence.
- **Seq2seq-LSTM-attention$_{\text{+charge\&law}}$:** this method uses the Seq2seq-LSTM-attention method described above, and also it includes the charge label and law article information in the decoder side.
- **DCA:** Celikyilmaz et al. (2018) use multiple encoders to represent the document together with a hierarchical attention mechanism for decoding. Their proposed Deep Communicating Agents (DCA) model is trained end-to-end with reinforcement learning.

Experimental Settings and Training.

Like previous studies, we use the validation data set to tune our model and hyper-parameters. The maximal length is 510 for both the fact descriptions and court views, although most court views in our data sets have less than 60 words. Model performance was checked on the validation set after every 300 batches training. Training process was terminated if the model performance is not improved for successive 10 times.

Encoder. We use the BERT-Chinese pre-trained model as the encoder for encoding fact description. This model uses 12 encoder layers, and the embedding size is 768 for the input token, the position embedding and the text segmentation embedding. The multi-head attention has12 heads, drop out is 0.1, L2 decay rate is 0.01, and activation function is Gaussian Error Linear Unit (GELU). The aggerated law article embedding is generated by the approach described before.

Decoder. The decoder side has six layers, as illustrated in Figure 2. The input is shifted one token position to the right, utilizing a teacher forcing learning approach. For the encoder-decoder multi-head attention in each of the six decoder layers, the inputs are the fact description embedding from the BERT encoder, the charge embedding, the aggerated law article embedding, and the output from the last decoder layer (or the court opinion input for the first layer), illustrated in Figure 3. The charge embedding size is 768 and is also randomly initialized. The other hype-parameters on the decoder side use the default values of the original Transformer architecture.

As described before, in our model, the BERT encoder and the decoder use two different optimizers. Their learning rates are different. We use two Adam optimizers with $\beta 1 = 0.9$ and $\beta 2 = 0.999$ for the encoder and the decoder, respectively, but they have different learning rates and warmup-steps. In Equation 2, we set $learnRate = 2e-3$ and $warmUp = 30{,}000$ for the BERT encoder, and we set $learnRate = 0.05$ and $warmUp = 15{,}000$ for the decoder. For the whole model, the batch size is 32 and dropout is 0.1.

Evaluation Results and Analysis

Experiment Result.

Table 2 and Table 3 present the comparison result on the two data sets. B-4 means BLEU-4 value. R-1, R-2, R-L are the F1 values of ROUGE-1, ROUGE-2 and ROUGE-L, respectively. The result shows that our proposed approach performs clearly better than all the other methods. To verify if the performance improvement is statistically significant, we conducted *t-test* between our model and the two best ones (i.e. DCA and SUMEXTABS). The *t-test* results show that the performance improvement is statistically significant at the level of *p=0.01* on both data sets, and for all the four evaluation metrics. The scores on data set 1 are lower than that on data set 2, and the reason is that the average length of court view in data set 1 are longer than data set 2, and data set 1 have a mix of criminal cases and civil cases, while most of the cases in data set 2 are about criminal case.

From the two tables we can also notice that even the Random$_{\text{+charge\&law}}$ method has a relatively high score, which means that the expressions of court view with the same charge label and law articles are similar, with many overlapped n-grams. We can also notice that the BM25$_{\text{+charge\&law}}$ method, which is based on retrieving the case having similar fact description and also having the same charge label and law articles, performs pretty well. The seq2seq model with attention mechanism performs better that the BM25$_{\text{+charge\&law}}$ model. Adding charge and law information to the Seq2seq-LSTM-attention model will improve the performance more.

Table 2: Algorithm comparison result on data set 1

Method	R-1	R-2	R-L	B-4
Random	24.1	5.1	23.6	5.2
Random$_{\text{+charge\&law}}$	52.3	27.3	47.8	23.4
BM25	61.3	41.4	58.1	37.8
BM25$_{\text{+charge\&law}}$	65.3	46.0	62.2	41.1
Seq2seq-LSTM-attention	66.9	48.1	63.6	41.8
Seq2seq-LSTM-attention $_{\text{+charge\&law}}$	68.4	50.1	65.2	42.9
DCA	69.7	51.4	66.8	44.6
Our approach	**71.7**	**53.2**	**68.5**	**46.7**

Table 3: Algorithm comparison result on data set 2

Method	R-1	R-2	R-L	B-4
Random	26.5	6.2	25.1	6.4
Random$_{\text{+charge\&law}}$	54.7	29.8	50.2	25.7
BM25	63.5	43.7	60.3	40.1
BM25$_{\text{+charge\&law}}$	67.8	48.2	64.5	43.4
Seq2seq-LSTM-attention	69.1	50.3	65.9	44.0
Seq2seq-LSTM-attention$_{\text{+charge\&law}}$	70.9	52.5	67.7	45.8
DCA	71.9	52.8	68.3	46.9
Our approach	**73.8**	**54.6**	**70.6**	**48.7**

Ablation Test.

The underlying ideas behind our model are the following two main hypotheses: 1. Adding charge and law article information in the decoding process will improve court view generation performance, since it will provide more charge-discrimination information and article related information. 2. Our generation model with BERT's pre-trained model as the encoder and utilizing two separate optimizers for encoder and decoder will improve the court view generation performance. The results in Table 2 and 3 have proven that these two features together make our model outperform all other approaches. In this ablation test, we want to see the effect of each feature. The ablation test was conducted on the data set 1, which is larger than data set 2 and has both civil and criminal cases. The results are presented in Table 4. The results show that removing either charge or law article information from the model will decrease its performance, which validates our hypothesis 1. If both of them are not included in our model, then it still outperforms the seq2seq-LSTM-attention model (result line 5 in Table 4 vs. result line 5 in Table 2), which demonstrates that our hypothesis 2 is valid. This test also shows that adding charge and law article embeddings to the final linear and softmax layer will improve the generation performance (result line 5 in Table 4), since it provides more context for decoder to choose the right token.

Table 4: Ablation test result using data set 1

Method	R-1	R-2	R-L	B-4
Our approach	71.7	53.2	68.5	46.7
- without charge	70.3	52.2	67.1	45.2
- without law	69.9	51.7	66.6	44.5
- without charge&law in the final output layer	70.5	52.4	67.4	45.4
- without charge & law	69.2	50.6	65.9	43.7

Length of Court Opinion.
We wanted to see the relation between the length of the court view and the model performance. Figure 6 shows how the BLEU-4 score changes with the court view length. At length of 15, we have the best performance, and the score decreases as the length grows, with score of 48 at length of 30 and going down to 32 at length of 50. This is not surprising, because when the text to be generated is long, it is hard to capture the correct source information when generating a token, not matter the model is based on Transformer or other seq2seq models.

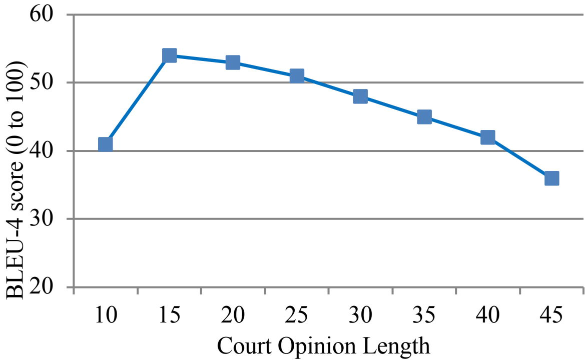


Figure 6: Model performance vs. court opinion length. The length is based on the number of tokens

Effect of Normalizing Numbers and Entities.
For numbers and entities appearing in facts description, unlike previous studies converting them to just two symbols, in this study, we differentiate the entities into four symbols and the converting numbers into 20 symbols, by analyzing the training data. We expect this change will make the fact description more distinctive for certain cases. By analyzing some examples, we found that this do make the generated court view more accurate. For example, there is a case about a government employee conducting the crime of corruption with a huge amount of money. We found that if we convert the names and numbers as other studies do, or do not do any conversion, the generated court view will have a BLEU-4 value of 0.33. In contrast, after the conversion using our approach, the score changes to 0.45. The reason is that without conversion, the view is generated as a crime of regular professional embezzlement, but with the conversion the generated view correctly tells that this is about corruption with public economy involving “数额特别巨大” (huge amount of money), and the applied laws and charge are also different.

Conclusion

In this study, we proposed a new method for automatically generating court opinion from the fact description of a legal case. Our model exploits the charge and law article information in the generation process, in addition to the fact description text. We use the BERT model as the encoder and a Transformer architecture as decoder. Our model employs two separate optimizers for training the encoder and decoder. The experiments on two data sets of Chinese legal cases show that our approach outperforms other approaches. One of the future studies in this direction will be to incorporate the predictions of charge labels and law articles into the court view generation process, which will be a jointly learning task.

References

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. Deep communicating agents for abstractive summarization. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

Ilias Chalkidis, Ion Androutsopoulos, Nikolaos Aletras, Neural Legal Judgment Prediction in English, ACL 2019

Yen-Liang Chen, Yi-hung Liu and Wu-liang Ho, a text mining approach to assist the general public retrieval of legal docuemnts. JASIST 2013

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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

Wenyu Du, Baocheng Li, Min Yang, Qiang Qu, Ying Shen, A Multi-Task Learning Approach for Answer Selection: A Study and a Chinese Law Dataset, AAAI 2019

Sergey Edunov, Alexei Baevski, and Michael Auli. Pre-trained language model representations for language generation. NAACL 2019

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut, ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, https://arxiv.org/abs/1909.11942, 2019

C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. NAACL 2003

Chao-Lin Liu and Chwen-Dar Hsieh. Exploring phrase-based classification of judicial documents for criminal charges in chinese. In Foundations of Intelligent Systems, 16th International Symposium, ISMIS 2006

Chao-Lin Liu and Ting-Ming Liao. Classifying criminal charges in chinese for web-based legal services. In Web Technologies Research and Development - APWeb 2005, 7th Asia-Pacific web conference.

Yang Liu and Mirella Lapata, Text Summarization with Pretrained Encoders, EMNLP 2019

Chao-Lin Liu, Cheng-Tsung Chang, and Jim-How Ho. Case instance generation and refinement forcase-based criminal summary judgments in Chinese, 2004

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, https://arxiv.org/abs/1907.11692

Yi-Hung Liu, Yen-Liang Chen, and Wu-Liang Ho. Predicting associated statutes for legal problems. IPM 2015

Daniel Locke, Guido Zuccon, A Test Collection for Evaluating Legal Case Law Search, SIGIR 2018

Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. Learning to predict charges for criminal cases with legal basis. EMNLP 2017

Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, Bing Xiang. Abstractiv text summarization using sequence-tosequence RNNs and beyond. CoNLL 2016

Ramesh Nallapati and Christopher D. Manning, Legal Docket-Entry Classification: Where Machine Learning stumbles, EMNLP 2008

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Ranking sentences for extractive summarization with reinforcement learning. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1, pages 1747–1759, New Orleans, Louisiana

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer, Deep contextualized word representations, NAACL 2018

Kishore Papineni, Salim Roukos, ToddWard, andWei- Jing Zhu. Bleu: a method for automatic evaluation of machine translation. ACL 2002

Romain Paulus, Caiming Xiong, Richard Socher, A Deep Reinforced Model for Abstractive Summarization, 2017, https://arxiv.org/abs/1705.04304

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019

K. Raghav, P. K. Reddy, and V. B. Reddy. Analyzing the extraction of relevant legal judgments using paragraph-level and citation information. In AI4JArtificial Intelligence for Justice.

Stephen Robertson and Hugo Zaragoza, The Probabilistic Relevance Framework: BM25 and Beyond, Foundations and Trends in Information Retrieval 2009

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. Leveraging pre-trained checkpoints for sequence generation tasks. arXiv preprint arXiv:1907.12461, 2019

Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. EMNLP 2015

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer generator networks. ACL 2017

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, Tie-Yan Liu, MASS: Masked Sequence to Sequence Pre-training for Language Generation, https://arxiv.org/pdf/1905.02450.pdf, 2019

lya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. NIPS 2014

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. NIPS 2017

Pengfei Wang, Ze Yang, Shuzi Niu, Yongfeng Zhang, Lei Zhang, Shaozhang Niu, Modeling Dynamic Pairwise Attention for Crime Classification over Legal Articles, SIGIR 2018

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le, XLNet: Generalized Autoregressive Pretraining for Language Understanding, NIPS 2019

Wenmian Yang, Weijia Jia, Xiaojie Zhou and Yutao Luo, Legal Judgment Prediction via Multi-Perspective Bi-Feedback Network, IJCAI 2019

Hai Ye, Xin Jiang, Zhunchen Luo, Wenhan Chao, Interpretable Charge Predictions for Criminal Cases: Learning to Generate Court Views from Fact Descriptions, NAACL 2018

Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liuy, Maosong Sun Legal Judgment Prediction via Topological Learning, EMNLP 2018

Xin Zhou, Yating Zhang, Xiaozhong Liu, Changlong Sun, Luo Si, Legal Intelligence for E-commerce: Multi-task Learning by Leveraging Multiview Dispute Representation, SIGIR 2019