Hindawi
Scientific Programming
Volume 2021, Article ID 3225933, 14 pages
https://doi.org/10.1155/2021/3225933

Review Article

Exploration of Cross-Modal Text Generation Methods in Smart Justice

Yangqianhui Zhang

The University of British Columbia, School of Biomedical Engineering, Vancouver, Canada

Correspondence should be addressed to Yangqianhui Zhang; arielzhang2018@alumni.ubc.ca

Received 7 May 2021; Revised 9 August 2021; Accepted 22 September 2021; Published 21 October 2021

Academic Editor: Liang Zou

With the development of modern science and technology, information technology has brought great changes to many fields. Smart justice has become one of the increasing areas that people are paying more attention to. For example, large and small cases occur every day, and the legal library is continuously updated. Therefore, a large number of documents and evidence collection archives will bring tremendous pressure on the judiciary. The text generation technology can automatically present the results extracted from these redundant legal data and express the results of the analysis in natural language. It facilitates the business for huge amounts of legal data effectively, which relieves the work pressure of the judicial department. However, the text generation algorithms have not been promoted in justice. Therefore, this paper focuses on what benefits text generation can produce in law and how to apply text generation technology in legal field. The survey provides a comprehensive overview on text generation firstly, through summarizing the existing methods, that is, text to text, data to text, and visual to text. Then, we examine the process of the practical application of text generation in law. Furthermore, this paper puts forward the challenges and possible solutions to the judicial text generation, which provides pointers on future work.

1. Introduction

For a country, law maintains social stability. For each individual, law is a powerful weapon to defend people's rights and interests. As a result, the work of the legal sector is often arduous and onerous. According to statistics, the legal database has collected nearly one million pieces of provisions. It is conceivable that judges cannot memorize all laws and regulations, thus affecting the fairness and efficiency of judgments. In addition, in recent years, Chinese citizens have visited, consulted, and handled affairs on the website of the Ministry of Justice hundreds of millions of times, which indicates that the legal department needs to devote a lot of time, manpower, and material resources to solve people's problems. Text data processing is particularly important in many judicial services. Automatic generation of legal texts can alleviate the shortage of legal professionals. Through the automatic generation of legal texts, the paperwork of legal service personnel can be reduced, thus improving the efficiency of generating legal documents and avoiding the waste of judicial resources. With the gradual improvement of the society ruled by law, the requirements of judicial activities in China are getting higher and higher, so the generation of legal texts is of great significance to the judicial field.

Automatic text generation is a technique in which a computer generates natural language from some form of data content. Natural language generation technology rose since the 70s [1]. The template generation (template-based generation) is the first use of text automatic generation technology. After that, the schema generation technology (schema-based generation) and phrases planning technology (phrase/plan expansion) which are based on the theory of RST (Rhetorical Structure) and many other technologies gradually appeared.

There are quite a few frontier research works on legal text generation in NLP (natural language processing) and artificial intelligence fields. In recent years, there have been

some achievements and applications with international influence in this field. Text automatic generation is the main research direction in the field of natural language processing, and deep learning algorithms play an important role in the field of natural language processing. In recent years, more and more researchers have combined the technology with artificial intelligence, such as Microsoft's chatbot "Xiaobing," Headline's news robot "Zhang Xiaoming," and Tencent's "dream writer." At present, automatic text generation technology has been successively applied in entertainment, meteorology, medicine, news, and other fields [2–4].

However, the technology is not yet fully available in the judicial system, but the importance of text generation in law should not be underestimated. At present, judicial artificial intelligence can simply realize legal retrieval, document search, and so on. Besides, some intelligent legal software has been put into commercial use, in which text generation technology has made many contributions. For example, in the Competition on Legal Information Extraction in 2018, Tran et al. [5] used text generation technology to represent documents with abstracts and achieved the best performance. Then, they used the 2018 model as a pretrained phrase scoring model and lexical matching technology in the 2019 competition. The model combining text generation techniques performed well again in the legal case retrieval task. In commercial software products, Kira can be used to extract the terms of the contract; RAVN systems can efficiently summarize your legal documents; Lex Machina analyzes the historical data of the lawsuit for lawyers and generates a report; Lisa and Automio robots can generate agreements and legal documents based on questions and answers from users. These beneficial features are inseparable from text generation technology.

In addition, automatic text generation still has rosy prospect in the judicial system. If the automatic text generation technology is widely used in the law system, it will greatly improve the efficiency of the workflow of law, which is a promising opportunity. Examples include the following: (1) When people need legal advice or case inquiry, due to limited human and material resources, the human window may not be able to provide timely services. In the process of inquiry, there may be some questions that are too embarrassing to mention, which may lead to the ineffective and inaccurate progress of the case. In the process of solving problems, the staff may not be able to find the appropriate provisions in hundreds of thousands of legal provisions in a short time [6]. Intelligent dialog system based on text generation algorithms can solve the above problems. (2) At present, the public security department has presented "data police," which can make prediction and give warning according to police data [7]. Regular work reports are indispensable. Therefore, some content selection can be made on these data, and relevant reports can be generated automatically by text generation algorithm. (3) To meet the requirements of modern information management, text generation algorithm can convert traditional files in the form of picture and video into document format for storage. (4) In traditional sentencing, judges need to read a lot of documents, which requires a lot of time and energy. If the text generation technology is applied to extract and summarize the contents of files and indictments, it can not only save time but also realize transparent and fair handling of cases. The possible application of automatic text generation in law is not limited to this, but it can be seen that legal text generation is very promising.

This article aims to explore the necessity and possibility of automatic text generation in the judicial system. First, this article will classify and summarize the existing classic text generation algorithms from three different forms of input content: text input, data input, and visual input in Section 2. Then, we explain in detail how to apply these text generation algorithms to justice with the existing works and provide 6 authoritative and available legal datasets that can be used for text generation or other artificial intelligence tasks in Section 3. Section 4 analyzes the possible problems and feasible countermeasures in the application of text generation algorithm to justice, which gives new research directions for both text generation and intelligent law. At last, Section 5 concludes the paper.

2. Automatic Text Generation

According to different input, automatic text generation can be divided into three categories: text-to-text generation, data-to-text generation, and image-to-text generation [8]. Each technology here is extremely challenging, but with the rapid development of natural language generation technology and artificial intelligence, each technology has more detailed classification and cutting-edge application methods. Text-to-text generation is divided into text summarization and dialog system. Text summarization is divided into extraction and abstraction forms to express the central idea of the article. Dialog system is intended to generate the natural language of response and generates models in three modes: template-based, knowledge-based, and network-based. Data-to-text generation is mainly divided into two solutions: content selection and surface realization, and rule-based and data-based methods are adopted. For visual-to-text generation, image and video input are integrated into visual form input. It can be realized by template-based and network-based methods. In this section, we will discuss text generation techniques: text-to-text generation, data-to-text generation, and visual-to-text generation. Figure 1 gives an overview of the automatic text generation methods.

2.1. Text to Text. Text-to-text generation is a technology to convert the given text content into a new text. The process of this technology mainly includes text summarization, sentence compression, sentence fusion, and text retelling, which can be applied in the fields of information summarization, news writing, system dialog, and machine translation. This section mainly introduces text summary technology and intelligent dialog technology which can be employed to smart justice.

2.1.1. Text Summarization. Automatic text summarization utilizes computers to extract simple coherent text content from the original text, which can fully and accurately express

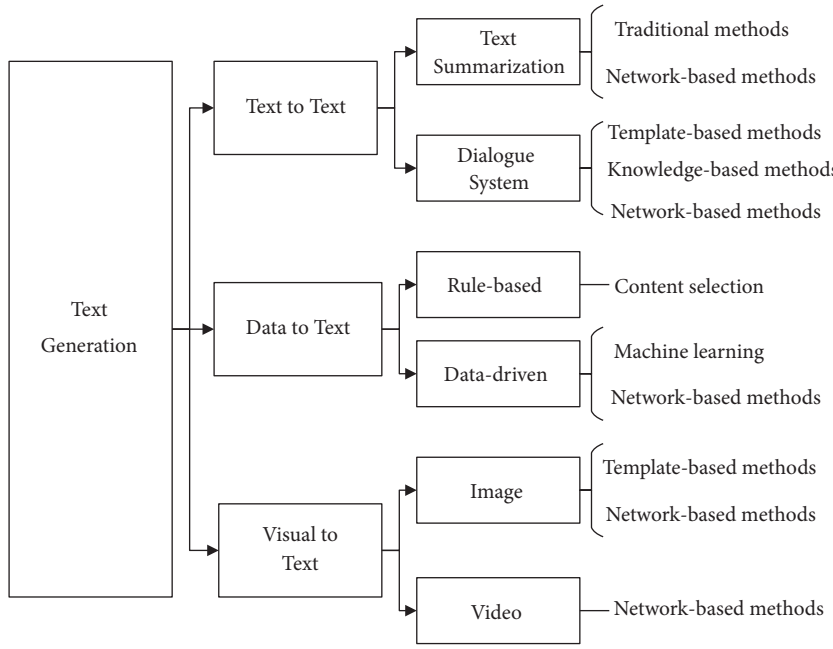

Figure 1: Text generation methods.

the central idea of the whole text. Summarization can be divided into extractive form and abstractive form: extractive form is composed of important sentences in the original text, while abstractive form is composed of new sentences. Traditional automatic text summarization is in extractive form.

In the original text summary methods, sentences were rated, sorted, and selected by word frequency, sentence position (first and last sentence), and keywords. Luhn proposed [9] to rate sentences according to the word frequency. The sentences with more frequent words have higher scores, and the sentences with higher final scores constitute the abstract of the text. This seemingly simple method sometimes has better effects than some complex methods [10]. In [11], Edmundson calculated the score of each sentence by integrating factors such as clue words, title, sentences at the beginning and end of paragraphs, and keyword frequency and selected sentences with high scores to form the abstract.

At the end of the twentieth century, machine learning emerged in text automatic summarization, making the process of summarization more intelligent. Inspired by Edmundson's idea, Kupiec added naive Bayesian classification model [12] to determine whether the extracted sentences meet the requirements of abstract. In 1999, Lin et al. applied the decision tree to the process of grading sentences and extracted the sentences with the highest scores to form an abstract. After that, Osborne [13] proposed an automatic text summarization method with a better extraction effect than the naive Bayesian model, which was based on the log-linear model and considered the relationship between different features.

In the twenty-first century, the emergence of neural networks made a breakthrough in text summarization technology. Kageback et al. [14] proved that the network-based text summarization method was significantly superior to other traditional methods. The automatic text summarization based on neural networks could generate the summarization of extractive form mentioned above or abstractive form [15]. The models can be divided into extraction model and abstraction model [16]. Among them, CNNs (convolutional neural networks) and RNNs (circular neural networks) were commonly used for neural-based abstracts, which were the basic models of many new technologies.

The extraction models focus on how to express sentences and how to choose the most suitable sentences. For example, CNNLM [17] employed convolutional neural network to represent sentences. Through training with noise contrast estimation, it can distinguish the real next word from the noisy word and select sentences based on the principle of optimizing submodule targets. This model can well process redundant information in candidate words. In [18], the method NN-SE utilized CNN and RNN to represent a sentence, which was input into the LSTM encoder. Thus, the LSTM decoder with sigmoid was used in grading, sorting, and choice of a sentence. Regarding the encoder and decoder, the stochastic gradient descent method was employed to minimize the negative logarithm likelihood. The contribution of this model is that the generation of abstracts no longer requires the manual language annotation process.

In [19], the SummaRuNNer was proposed to use a two-layer bidirectional RNN to represent sentences and documents, each of which was a bidirectional GRU. The model SummaRuNNer is shown in Figure 2. The blue part is the word-level representation, and the red part is the sentence-level representation. For each sentence representation, there is a 0, 1 label output indicating whether or not each sentence belongs to the summary. The second layer merges the sentence representation of the first layer into a document representation, in which the sentences are sorted using the

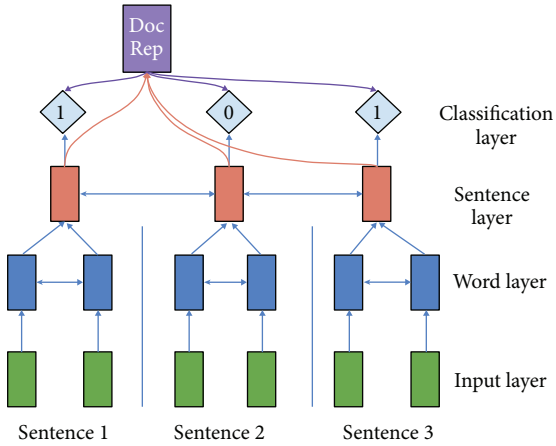

Figure 2: SummaRuNNer model.

sigmoid function. The training of this model is similar to the NN-SE model. Its advantage lies in the visualization of prediction, which makes the model intuitive and easy to understand. Moreover, its performance is comparable to that of some advanced depth models.

In the abstract summary model, the main consideration is how to represent the whole document in the encoder and how to generate sequence words through the decoder. For example, RAS-LSTM and RAS-Elman [20] used the encoder based on CNN and attention mechanism and used Elman RNN or LSTM model for decoding. The novel convolutional attention encoder of this model can ensure that the generation process of the decoder always focuses on the appropriate word input. Nallapati et al. [21] proposed a feature-rich hierarchical attention encoder based on two-way GRU to represent documents, in which one-way GRU, decoder based on LVT (the large vocabulary trick), and pointer switch mechanism were utilized. The innovation of this model is to model basic structures such as keywords, rare words, and word-to-sentence hierarchy, which will help improve the performance of the model. In [22], Pointer-Generator Networks adopted single-layer bidirectional LSTM as encoder and single-layer unidirectional LSTM as basic decoder and added pointer switch mechanism. On this basis, an overlay mechanism to punish repeated attention was also proposed. This model effectively solves the problem that the traditional sequence-to-sequence neural network model is prone to duplicate inaccurate content [23–25].

2.1.2. Intelligent Dialog. Automatic text generation in the dialog system refers to the natural language of the organization to generate responses based on the user's statement. Intelligent dialog system currently has three modes [26]: template-based, knowledge-based, and deep learning-based sequence-to-sequence generation model.

(1) Template-Based Models. This technique designs dialog templates for specific scenarios, and the text generation process is a template filling process [27]. The template-based model can accurately answer the questions in a certain field, but it has poor portability. It is suitable for the scenario of human assistant.

Apple's Siri uses template-based natural language generation. Siri employs the system's vocabulary to map surface words to related concepts, relationships, and properties, creating a dialog template that allows it to interact easily with users.

(2) Knowledge-Based Models. Based on an indexed dialog database, the user's statements are first analyzed using natural language processing (NLP) technology, and then fuzzy matching is performed in the statement database to select the response statements with the highest matching degree. This model is often used in entertainment chat and question-and-answer systems, and its knowledge base is easy to expand. However, when the amount of data is too large, the context is often not connected.

IBM's computerized question answering system, Watson, uses knowledge-based retrieval technology during the text generation stage [28]. After collecting large-scale evidence, Watson further analyzes and evaluates the answers. The system uses Deep QA architecture, which follows: (1) including more than one principle of assertion for the answers of fact, (2) searching for different resources for different understandings of the problem [29], and (3) achieving more than one candidate answer. After evaluation, scoring of each answer, the best answer is finally selected. Moreover, the complementarity of unstructured information and structured information is employed to improve the correctness of evidence analysis [30].

The architecture of Deep QA is extensible, in which Q&A tasks can be improved through the expansion of the knowledge base. However, the knowledge base is growing too fast to be updated in real time.

(3) Deep Learning-Based Models. Dialog generation based on deep learning does not rely on any template or knowledge base. This model is based on the end-to-end technology of deep learning, which acquires the ability of organization by learning natural language directly through a large amount of corpus. The resulting text is more flexible and intelligent.

Google [31] proposed a sequence-to-sequence framework to train their conversation engines. The model used end-to-end training patterns and backpropagation learning. The output of the conversation was based on the predicted sentences or sentences in the conversation. The completely data-driven approach can save a lot of manual overhead, but the model is capable of only simple conversations and lacks consistency before and after conversations.

Sordoni et al. [32] added context relation on the basis of the previous model and replaced the RNN model with multilayer forward neural network, so that the model could input context information and dialog information into encoder and maintain the dynamic consistency of input and output information. This context-aware approach also presents problems, such as adding distant content unnecessarily to the current generation process.

Kumar et al. [33] proposed a model of dynamic memory network, which used an episodic memory module to store context information and corpus based on HNN. Dialog is a

process of iterative attention; thus, the final text generation will be a hierarchical recursive reorder, resulting in a high-quality dialog generation.

It can be seen that neural network performs well in both text summarization and conversational system technology. However, when generating real sentences, there is a high probability of failure for two main reasons: (1) When using autoencoders to map sentences to their hidden representations, the representations of these sentences often occupy a small area of the hidden space. Therefore, most areas in the hidden space are not necessarily mapped to real sentences [34]. (2) Because of the nature of RNN itself, the error rate of sentence generation may increase greatly with the length of the sentence itself, which makes the quality of long sentences difficult to be guaranteed. In order to solve the above problems, in recent years, researchers pay more attention to how to generate more realistic sentences; they usually adopt the following methods: (1) using GANs (Generative Adversarial Networks) [35] frame to make the text more like human writing; (2) using reinforcement learning; (3) combining semantic or grammatical information to make the resulting sentences more correct [36].

The application of adversarial training can effectively improve the above problems by alternately updating discriminator and generator. Zhang et al. [34] proposed a method of adversarial training texts, which utilized LSTM as a generator and CNN as a discriminator. The generator constantly generates near-real sentences, and the discriminator aims to accurately distinguish the sentences generated by the generator from the real sentences. After adversarial training, the sentence was guaranteed to maintain high quality from a holistic perspective. In addition, Li et al. [37] have applied adversarial training to the neural dialog system, making the dialog generated by the intelligent dialog system almost indistinguishable from human language.

In 2019, Gao et al. [38] applied a GAN model to add text-related comment information. They chose the Seq2Seq model based on the attention mechanism and pointer mechanism as the generator and CNN as the discriminator. They used the content of the comments to get the main ideas and redundant information in the text.

Zhang et al. [39] used a more powerful generator, Transformer. Transformer is a completely attention-based model proposed by Google in 2017. It has achieved excellent performance in machine translation. Similarly, they chose CNN as the discriminator. The efficient parallelization of the Transformer framework has made their work a good result.

The GAN model is often combined with reinforcement learning. The GAN model alone cannot be applied to natural language generation, because the generated data of text is discrete, and the improvement of generator is effective for continuous data such as image based on discriminator information. However, for text data, the improved results are likely to correspond to invalid text information. Reinforcement learning has an inherent advantage in discrete data. It can use customized reward or punishment mechanisms to drive the final result more flexibly. Therefore, GAN-based text generation models usually use the policy gradient method in reinforcement learning during the training of the generator and discriminator. The above models of Gao et al. [38] and Zhang et al. [39] were designed as such. Of course, reinforcement learning itself makes a great contribution to natural language generation. Chen and Bansal [40] first used a deep learning model to extract important sentences and then used reinforcement learning to abstract the extracted text. Their model uses the idea of parallel decoding, which makes the decoding process very efficient.

In the dialog system, the process of dialog is like a decision-making process, so it can be fitted by the strategy learning process of reinforcement learning. Li et al. [41] used adversarial inverse reinforcement learning technology and provided a unique reward mechanism for the discriminator of the adversarial model, so the generator can obtain more accurate reward signals from it. Experiments have shown that their dialog system can produce high-performance responses.

In addition to the above two advanced methods, it is also an effective and feasible way to return to the semantic and grammatical structure of the text. Kouris et al. [42] proposed a new model combining deep learning and semantic data transformation in 2019. The principle of conversion is as follows: generalize the low-frequency words in the text into high-frequency words in the learning process, and then materialize the words in postprocessing. Based on this principle, the prediction of the model becomes more accurate.

Song et al. [43] expressed text as Abstract Meaning Representation (AMR) [44], which describes the grammatical structure of a sentence. They used a novel graph-to-text encoder. The traditional graph-to-text method is to traverse the nodes in the graph in a depth-first search or a breadth-first search. This method has disadvantages, which will cause the words that are close to each other to be far apart after traversal. Therefore, they use RNN to directly encode the graph into text, which solves the above problem well, and this parallel encoding method saves a lot of time.

The accuracy of text-to-text generation model still needs to be improved. The training of a text model usually requires a large corpus, and the training time of the model is very long (it may take several days). Therefore, an efficient text generation model is needed. Recently, many scholars have tried joint training on multiple documents. Fabbri et al. [45] applied a single-document model to multidocument text and found that combining methods such as Maximum Marginal Relevance (MMR) [46] is feasible. However, they just simply connect multiple documents, and the relationship between different documents is not considered. Effective multi-document-based text generation is also an important research direction in the future.

2.2. Data to Text. The generation of data to text is based on various data and tables, from which the internal structure and correlation are analyzed to form a smooth text. Ehud Reiter of the University of Aberdeen put forward the general framework of the data-to-text generation system [47], as shown in Figure 3. Firstly, the numerical data is input from the signal analysis module, and the basic patterns in the data

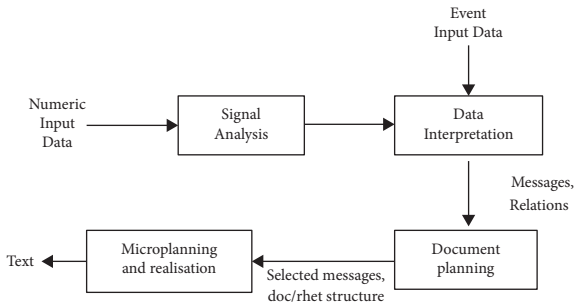

Figure 3: Data-to-text generation system.

are detected by various data analysis methods, which are output as discrete data patterns. The input of the data interpretation module is the basic events. By analyzing the basic patterns and input events, more complex and abstract messages are inferred, and their relationship is inferred. Finally, the high-level messages and the relationship between messages are output. Then, enter messages and relationships in the Document Planning module, analyze and decide which messages and relationships need to be mentioned in the text, at the same time determine the structure of the text, and finally output the messages and document structures that need to be mentioned. The last step is to input the selected message and structure in the Microplanning and Realization module and output the final text through natural language generation.

At present, this technology is mainly used in the fields of meteorological report, finance, sports, and medicine. There are two main problems that need to be solved in data-to-text generation: (1) how to choose effective data subset from the data obtained, which can be called content selection; (2) how to describe these data subsets in human language, which can be called surface implementation. The actual methods of data-to-text generation can be divided into rule-based methods and data-driven methods [48]. The relevant development context and research methods are introduced as follows.

2.2.1. Rule-Based Methods. Rule-based data text generation methods make the content selection and natural language representation of data according to expert knowledge in a certain field, which is suitable for specific fields, such as medicine and meteorology [49].

In the medical field, Hallett et al. [50] proposed a medical research method based on medical history information in 2006. The innovation of this method is to encode the information of clinical history into data and use the medical history data to generate text reports to support further clinical research. The approach also incorporates visual navigation tools to address the shortcomings of text generation. It aims to study cancer-related problems, but the method could be applied to other areas of medicine as well. In 2009, Gatt et al. proposed [51] "BabyTalk" system to generate the natural language summary of neonatal intensive care data. This system adopted the algorithm proposed in [50] to combine data with visualization and other technologies, to make the decision-making results more accurate.

Banaee et al. [52] developed a text generation method based on physiological sensor data in 2013. This method extracts information from the original data, performs data denoising and other processing, and uses expert knowledge to delete the value of the workpiece, to ensure that the system can generate text according to reliable signal input. In the natural language generation stage, the system uses correlation functions to order the importance of sentences and finally outputs robust text.

In the field of meteorology, Ramos et al. [53] proposed a meteorological service system "GALiWeather" in 2014, which took the weather data as the initial input and abstracted the data values into time-related language labels, namely, an intermediate code, through a computational method. Finally, the intermediate code was used as secondary input to generate natural language descriptions using an NLG system containing expert rules. They designed two NLG systems: One dealt with simple variables (cloud cover, wind, and temperature), in which language templates were defined. The other dealt with precipitation variables to prevent repetition, redundancy in the generated sentences. This method can guarantee high performance in content and form of text generation and can generate text description close to expert generation. However, the system is currently only applicable to the field of meteorology, with poor universality.

In 2016, Gkatzia et al. [54] developed two natural language generation systems, one based on "WMO (world meteorological organization)" and the other based on "NATURAL." Both systems provided text descriptions of precipitation and temperature, improving the accuracy of prediction. WMO is a rules-based system that can make predictions such as a 30 percent probability of rain, taking into account an interval of sunny days. The system can then generate the following text description: "it may be sunny, it may be rainy–less likely than impossible." The NATURAL system can imitate the tone and description of a weather forecaster. The rules used in this system come from the way in which observations (such as the BBC weather reporter) make predictions. For the same example above, the system obtains the following text description: "mainly dry and sunny."

The above methods can demonstrate that the rule-based data text generation needs the power of experts, and it can perform well in professional fields, but the applicability of the model is not wide. Moreover, rule-based methods often require a language template, which makes the generated text form too monotonous. Fortunately, the data-driven approach can improve both of these problems.

2.2.2. Data-Driven Methods. Data-driven text generation refers to the direct use of data for training, without the intervention of expert knowledge [49]. At present, data-driven methods have dominated natural language generation.

Liang et al. proposed a probabilistic generation model in 2009 [55], which can uniformly deal with the correspondence from segmentation text to description, fact identification, and data-to-text matching and solve the increasing ambiguity and noise in data. Inspired by this, Angeli et al. designed a new log-linear classifier in 2010 [56]. The whole text generation process is decomposed into several local decisions, which proved to have high performance in different fields such as sports and weather.

In 2014, Sowdaboina et al. proposed to utilize machine learning (ML) technologies to solve the problem of data content selection for the first time [57]. The model uses a mixture of natural language generation techniques and template-based methods to help the NLG system select text suitable for the application of templates, thus combining their respective strengths to produce high-quality text. The use of machine learning makes the rules of text generation closer to the human mind.

In the same year, Gkatzia et al. [58] introduced the feedback mechanism based on the content selection model in [57]. They compared and discussed the methods of multilabel classification and reinforcement learning (RL). The results showed that ML technologies can make the prediction results more accurate, while reinforcement learning is more exploratory.

In recent years, deep learning has achieved remarkable results in text summarization technology, and it also performs well in data-driven text generation. Mei et al. [59] proposed an end-to-end neural network model in 2016, which does not require the intervention of experts or rules. The model uses an encoder-allocator-decoder architecture and employs LSTM network unit as nonlinear encoder and decoder. In the model, the bidirectional LSTM-RNN encoder takes input from a set of event records and obtains the representation after modeling the dependencies that exist between the records in the database. The aligner of the model performs content selection using an extension of the alignment mechanism. This model can achieve satisfactory results even in fields where data is scarce.

In 2016, Lebret et al. [60] introduced a feedforward neural language model based on conditional neural language models, which can regulate text generation by tabular conditional language model and generate the sentences of people's biographies according to the fact tables in the dataset of people's biographies in Wikipedia. It copies and transfers words from fixed vocabularies and sample tables into output statements, which is a way to process large vocabulary data. The model has a good grasp of the tenses of the text, but some words need to be correctly predicted under a global condition. Overall, the model is able to generate fluent one-sentence descriptions of each character. However, generating longer descriptions is the problem that they have to tackle.

In 2019, Liu et al. [61] layered reinforcement learning frameworks to accommodate multimodal tasks. The model consists of multilevel strategy mechanism and multilevel reward mechanism. The first part aims to improve the accuracy of word level and sentence level, since the multilevel policy network can adaptively integrate word-level and sentence-level policies to generate each word. The second part guides the reward mechanism by combining image and language information. In order to better connect policies and rewards, they also designed novel optimization guidance items [61], as shown in Figure 4.

The difficulties of data-driven methods are mainly as follows: (1) There are high requirements for reliability and accuracy of data sources, which will directly affect the accuracy of the generated text. When dealing with large-scale data, the performance of the model decreases dramatically. (2) Efficiency is low when facing large-scale data. Wiseman et al. [48] employed a series of advanced neural methods and a simple template generation system to tackle document generation tasks. Experiments have shown that recent neural network models perform well in generating short textual descriptions of small amounts of data, but when faced with large-scale data, even with the ability to generate smooth text, text descriptions and human-generated documents still have a large difference. Puduppully et al. [62] found that if the content planning of data is carried out in advance, it can make a good combination of a large amount of data and deep learning model. They identified two questions before modeling to specify what to say and in what order. Experiments proved that they made a correct attempt, and the generated text had better conciseness and grammar. However, this method only improves the overall quality of the text, and more research is needed to accurately express the details. At present, there are not many generation models for large-scale data, so how to overcome the challenge brought by massive data is still a serious problem. To further advance data-driven text generation, both the bottlenecks must be addressed.

2.3. Visual to Text. With the popularization of all kinds of electronic products and the development of multimedia technology, a large quantity of pictures and video information is generated every day. If the multimedia information is accurately converted into descriptive text, the efficiency of classification and management can be greatly improved. The text generation work of image and video is rough as follows.

2.3.1. Image to Text. Image-to-text generation is a natural language description process after analyzing the visual content of image. There are two main ways to generate text from images: (1) The text can be generated through predefined generic language templates, during which the key attributes of images and other effective information are added. (2) Deep learning researchers generate descriptive sentences by using sequential generation models. These two generation methods are described below.

(1) Template-Based Generation. The template-based methods first use computer vision technology to identify the objects in the image, preset the template to be filled, and populate object relations and attribute labels into the template to generate the descriptive language of the image.

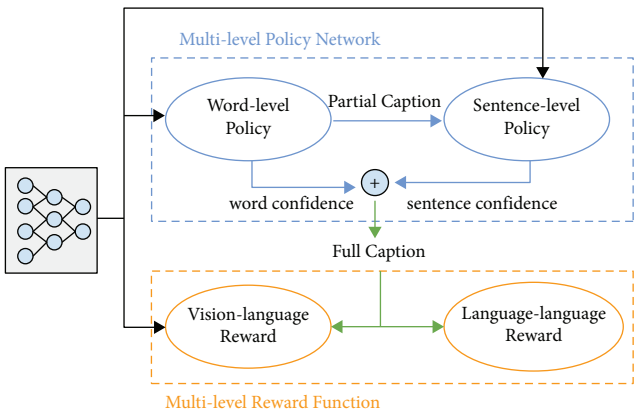

Figure 4: Multilevel policy and reward.

Farhadi et al. [63] first proposed the idea of cross-modal transformation from picture to text and studied the method based on language template. The model assumes that there are three spaces: image space, sentence space, and meaning space between them. The model uses triples (object, action, scene) for meaning representation. For sentence space, they use Curran & Clark parser [64] to generate the dependencies of each sentence and extract the subject-verb-object and other structures of the sentence, and then add them to the template of the sentence. By learning the mapping of image space and sentence space to the meaning space, measuring the similarity between them, and establishing the connection with the meaning space, the two-way conversion of image and text can be realized [65].

Kuznetsova et al. [66] proposed a new tree-based template method, which generated tree-structured phrase fragments by learning existing training sets, and then selectively combined these fragments to generate text descriptions. This model has a stronger generalization and generation ability than previous methods.

Yang et al. [67] utilized the hidden Markov model in the template-based method to fill in the template of sentence generation with the most likely predicted subject-object, verb, preposition, and other contents and finally output the natural language description of the image after decoding. The sentences generated by this model are more readable and relevant, but sometimes they are less predictive of nouns and verbs.

Language template-based methods tend to produce monotonous sentence patterns and content. In order to solve this limitation, deep learning-based coding-decoding methods are a better choice.

(2) Deep Learning-Based Generation. The implementation process of deep learning-based coding-decoding methods is naturally divided into two parts: The coding process is designed to extract visual features, generally using deep neural network, CNN. In the decoding process, extracted features are used as input, and natural text describing image is generated by using RNN or LSTM model. Coding-decoding methods are the applications of deep learning in image text generation, which often combine some different fusion methods, attention mechanism, or reinforcement learning [67] to generate more diverse sentences.

Mao et al. [68] first proposed an image text generation model m-RNN based on neural network. In addition to the CNN-based visual feature extraction part and language modeling part, this model also has a multimodal part, which connects the language model and CNN through a layer representation in m-RNN. m-RNN model can not only complete the image-to-text generation, but also solve the problem of sentence and image retrieval.

Fang et al. [69] proposed a new image generation text model, which consisted of three main parts: (1) visual detector, used to identify high-frequency words in image titles; (2) language model, adopting the CNN structure, which is used for the statistics of the related information of words and the generation of natural language; (3) a multimodal similarity model, which is for reordering words. The model is directly studied in the title text of the image, which combines the image content to obtain words of various parts of speech, ensuring that the generated text contains these words. Its global semantic result is the best in the official benchmark test.

Xu et al. [70] added attention mechanism to the model. It used convolutional neural network as an encoder to extract feature vectors of images and used long and short time memory networks in the decoder. The generated position of each word was determined according to the context vector, past hidden state, and position of previously generated words. The mechanism of the attention model allows the algorithm to selectively focus on certain areas of the image, thus visually selecting important parts.

Zhou et al. [71] proposed a special attention-based approach, which focused on words in the text, as opposed to the traditional approach focusing on part of the image. The model uses td-gLSTM (time-dependent gLSTM) method to generate attention guidance signal, which guides LSTM to generate descriptive natural language.

In recent years, reinforcement learning has become a hot topic in machine learning. In 2017, Zhang et al. [72] applied reinforcement learning to the process of image text generation. The model uses actor-critic method to train, thus it can improve the matching between training results and prediction results through the mechanism of reward and punishment. In the same year, Ren et al. [73] developed a new decision-making framework, which used the "strategy network" and "value network" in reinforcement learning to generate texts collaboratively.

However, using only the reward and punishment mechanism in reinforcement learning and the strategy network to generate text images is still unsatisfactory. Multitask learning vision and language pose a challenge to generation. In 2019, Liu et al. [61] layered reinforcement learning frameworks to accommodate multimodal tasks. The model consists of a multilevel strategy mechanism and a multilevel reward mechanism. The first part aims to improve the accuracy of both the word level and the sentence level, and the second part guides the reward mechanism by combining images and language information. In order to better bridge strategies and rewards, they also designed a

novel optimization guidance item. Aiming at solving the problem of multimodal learning, Nguyen et al. [74] also added detailed natural language descriptions of objects based on title information and combined the mixed end-to-end CNN-LSTM model to effectively solve the two problems of natural language generation and object retrieval of object titles.

Although image-to-text generation methods are constantly being innovated, there are still many problems to be improved, such as the immature image feature extraction technology, the semantic gap between image and text, and the cross-language description of images [75].

2.3.2. Video to Text. Early video-to-text generation works depended on the manual operation of the video feature extraction and modeling tasks [76, 77]. After that, more and more research was proposed. In 2015, Xu et al. [78] designed a new discriminative CNN to learn video representation for event detection. However, this model ignores the time structure of video. In order to solve the problem, Ballas et al. [79] proposed GRU-RCN algorithm, which considered video time and space feature information. It can obtain more refined video motion information in order to reduce the bad influence brought by high-dimensional video reproduction.

Pan et al. [80] also proposed a hierarchical recursive neural encoder (HRNE) to generate text for video, aiming at the integration of time information in video. The hierarchical structure enables video information to be better expressed, and the higher part of the model can make full use of the time structure and can be transformed at different granularity of time. In addition, the HRNE model has promising flexibility and nonlinearity, but the generalization ability of the model needs to be improved.

The above models are only used to generate a few sentences of short video. Yu et al. [81] first attempted to use deep learning method to generate multiple sets of statements or paragraphs for long video in 2016. They proposed a framework based on RNN structure to generate video paragraph text. The framework consists of a sentence generator and a paragraph generator. The paragraph generator models the relationships of simple sentences generated by the sentence generator. This algorithm has achieved favorable results in two large datasets, YouTubeClips and TACoS-MultiLevel, but the model is unable to process very small objects in video. Besides, the error superposition may occur due to the unilateral nature of the sentences generated by the model. All these problems need to be solved.

For video's multimodal features, many current models simply connect the features of video with different modes. Xu et al. [82] focused on the characteristics of video's multimodal features and proposed a multimodal attention span memory neural network (MA-LSTM) model. LSTM encoders and decoders are used in the model. Because video has multimodal characteristics, three LSTM models are built to encode video frames, video motion, and audio, and then they are fused to form multimodal flows, which are then output from the decoder. A multilevel attention mechanism is added to enhance the flexibility and effectiveness of modal integration. Compared with the advanced video-to-text generation algorithms GUR-RCN and HRNE, this algorithm has more obvious advantages and is a more successful network model.

3. Application of Text Generation in Smart Justice

This section will discuss the practical application of text generation in justice with existing generation models. Prior to this, we will introduce 6 authoritative legal datasets that can be used for text generation, such as judgment prediction and clerical generation. Of course, they can also be used for other intelligent judicial tasks.

3.1. Legal Case Reports Dataset. (https://archive.ics.uci.edu/ml/datasets/Legal+Case+Reports)The dataset was provided by the Federal Court of Australia (FCA). It includes all the legal cases of the Federal Court from 2006 to 2009. For each document, the dataset contains its catchphrases, citations sentences, citation catchphrases, and citation classes. These data can be used for automatic text summarization and citation analysis.

3.2. Department of Justice Open Data. (https://www.justice.gov/open/open-data) US Department of Justice published a list of legal data publicly online on November 30, 2013, so this dataset is a high-quality open dataset. It includes specific databases such as violent crime cases, FBI crime reports, and statistical reports.

3.3. The Supreme Court Database. (http://scdb.wustl.edu/) The database comes from the US Supreme Court and has absolute authority. The data records cases of court judgments from 1791 to 2018. Each case contains the legal provisions referenced by the case and many details at the time of the decision.

3.4. Caselaw Access Project (CAP). (https://case.law/) The database contains 360 years of various judgment cases in the United States, which have been digitally obtained from the collections of the Harvard Law Library. The cases have been organized into a unified form. A total of 1,693,904 different cases have been collected.

3.5. Bureau of Justice. (https://www.bjs.gov/index.cfm?ty=dca) The data source is provided by the Bureau of Justice Statistics and contains data on some US law enforcement agencies, prisons, parole, and probation. This data is essential to improve the efficiency of legal offices and effectively help fight crime.

3.6. CAIL2018. (https://github.com/thunlp/CAIL2018) CAIL2018 [83], the first large-scale legal dataset for judgment prediction in China, is derived from the website of adjudication documents. The dataset includes 2676,075 legal cases, all published by the Supreme People's Court. Each

case includes a description of the facts of the case and the outcome of the judgment, which is embodied in the relevant legal provisions, the predicted charges, and the sentence. This dataset is very large and very well annotated.

It can be found that most of these legal datasets are composed of text, so text-to-text generation technology plays a vital role in the generation of legal texts, which is also the current research content of most researchers. However, the potential contribution of data-to-text and visual-to-text technologies to legal work cannot be ignored.

3.7. Application of Text-to-Text Generation. Automatic text-to-text generation technology can be applied to intelligent extraction and intelligent dialog in smart justice. The application of text summary technology to the reading and summary of case documents can relieve the pressure of judges and reduce the errors caused by human operations. For example, in order to better solve the issue of appealing for disability benefits for veterans, Zhong et al. [84] hope to extract important sentences from cases as summarization. The abstracts can help the Board of Veterans' Appeals (BVA) to make more accurate decisions on cases. They used a corpus of about 35,000 BVA cases on disability compensation for posttraumatic stress disorder (PTSD). The authors used the idea of train-attribute-mask pipeline, sentence type classifier, and MMR technology successively to select summary sentences with a priority prediction function and finally embedded the generated sentences into a template. The selection of an advanced abstract neural network model is the key step for intelligent extraction. During the generation of abstract, additional modeling or attention can be paid to such important information as time and place.

The retrieval model based on judicial knowledge base can be used in the intelligent dialog system of judicial domain. Firstly, a complete judicial law knowledge database is built, which can be expanded or deleted according to the modification of laws and regulations, and the appropriate algorithms are selected to evaluate the matching statements. Finally, the response statements are generated. Governatori et al. [85] extended an existing dialog framework into the legal field. They used the framework to model the process of dialog in legislative deliberations. For more flexible questions, deep learning-based dialog system can be used to answer.

3.8. Application of Data-to-Text Generation. Data-to-text automatic generation technology can be applied to intelligent report generation in smart justice. Usually, the legal system creates files for each criminal, records the occurrence of some cases, etc. Thus, we can establish a database of this content and make corresponding structural selection. For example, using one kind of criminal event or a certain period of time of the case records, the natural language description can be generated automatically based on data-driven text generation algorithms.

GAN is improved by Kang et al. [86]. The encoder-decoder model based on LSTMs is used as the generator, and the binary classification module based on CNN is used as the discriminator. Through the real legal documents of divorce cases and through the data-driven method, a total of 25,000 case report datasets were preprocessed by word segmentation. Finally, through comparison, it is concluded that the text index of case description generated by this model has a good effect.

3.9. Application of Visual-to-Text Generation. Automatic visual-to-text generation technology can be applied to intelligent storage in smart justice. With the gradual informatization of legal systems, document storage format is no longer the traditional JPG, PNG, MPEG, MP4, and other forms of pictures and video; they need to be expressed into text. Image files can use the infrastructure of encoder and decoder in deep learning to generate natural language descriptions. Kang et al. [86] constructed a deep learning network model, ED-GAN, which is suitable for automatic generation of legal texts and applied the model to the generation of legal case description. At the same time, the discriminator model based on CNN can improve the accuracy of the generated text and form a competitive confrontation with the real text. The method can generate the case description text for a long time through the network-based method. The experimental results show that ED-GAN model has a good effect in generating case description text. If necessary, other technologies should be combined to enhance the learning ability of images. In judicial work, a video, which is monitored for a long time and has a lot of redundant information, is generally processed. Therefore, in addition to video-to-text generation model with good multimodal characteristics, attention mechanism is often needed.

4. Challenges of Text Generation in Smart Justice

In this section, we further discuss the text generation techniques according to the characteristics of judicial text and judicial work and locate the problems and challenges in their applications in judicial work.

(1) The text generation algorithms cannot yet be used to solve complex problems in smart justice. For example, in the existing intelligent consulting service, the intelligent dialog systems are realized by text-to-text generation. However, the existing dialog systems are not perfect enough. When dealing with complex problems, manual services are still needed. This indicates that the current natural language generation models are not fully capable of thinking like a human brain. We look forward to the day when computers can be answered like humans, which is not just simple and mechanical. However, this requires further development of artificial intelligence in text generation for smart justice.

(2) The performances of the existing techniques have not met the standards required by law. From the characteristics of judicial text, it is different from other

texts. In essence, the law is the highest standard of conduct used to regulate and constrain the whole society, which is formulated or recognized by the state and guaranteed by the state's coercive force. It has supreme authority and prescriptiveness. Concreteness, accuracy, simplicity, preciseness, and specification are the standards of wording in legal texts [87]. In the previous section, some neural network-based text generation algorithms were summarized. However, due to the inherent nature of the model, the high quality of sentences cannot be guaranteed when generating long sentences. Even if adversarial training was used, it can only improve the overall quality of the sentence.

To tackle this, the corpus should be accurately classified or extracted for keywords before sentence training, and the idea of keyword coverage should be used for modeling. Thus, promising results may be achieved. However, judicial text generation should improve its efficiency as much as possible in the links of input, training, and output. Therefore, the study of high-quality text generation technique is another promising area in smart justice.

(3) The generation of judicial text needs to standardize the wording and format. Specifically, legal terms have a single meaning [61], and each term represents a specific legal concept, which cannot be arbitrarily replaced when used. For example, "alimony" refers to "alimony for divorce," which cannot be replaced with "payment," even though in reality the terms are similar. Besides, legal terms also have opposite meanings [87]; namely, many terms come in pairs with contradictory meanings, such as plaintiff and defendant, actor and victim. Therefore, it is necessary to accurately grasp the subject and object in the text, and there must be no situation where the host and the guest are upside down [88].

In the generation of judicial texts, their characteristics should be fully considered, and the terms in corpus should be used accurately. A semantic-driven approach can be used to study judicial documents. This model should consider the complex structure and semantic knowledge of judicial texts to enhance the application effect in law. Besides, before using the text generation model, a domain knowledge model of judicial documents should be constructed. The more accurate the knowledge model is built, the more effective the results will be. Therefore, the construction of the domain knowledge model is pretty important in smart justice.

(4) The size of the data generated by judicial texts is huge. There are nearly hundreds of thousands of laws and regulations that need to be entered into the system. At present, tens of millions of judicial documents have been published. Different from meteorological and news fields, the storage of judicial data needs to be more complete and lasting, which proposes certain requirements for its storage technology. Moreover, the current text generation techniques are not good at large-scale data, especially in data-driven text generation.

Therefore, the text generation models should be combined with some advanced caching technologies to solve the storage problem of a large amount of text data. For example, the extension mechanism based on replication and reconstruction can effectively improve the neural network system of a large amount of data, but the overall efficiency is still limited. Thus, more research is needed to make a significant breakthrough.

(5) Models need interpretability. Many models of artificial intelligence are like black boxes, which may produce correct but abstract results. If the results of a model are not well explained, they may not be convincing, especially in the serious and infallible field of law. Keppens et al. [89] used Bayesian networks in legal decision making. Encouragingly, their model can well explain the production results, coupled with the probabilistic rationality of the Bayesian network. This will be an acceptable one in the legal field model.

Thus, if the text generation technology is combined with knowledge such as mathematical statistics or given a reasonable explanation for each process in machine learning, advanced models will be better promoted in law.

Text generation technology needs to integrate the research results of natural language processing, machine learning, cognitive science, and other fields, and it has very high research value and prospects. However, smart justice has great challenges in text generation because of its complex problems, strict wording standards, and huge data specifications. Therefore, in this case, this paper proposes a cross-modal legal text generation direction as a future research opportunity. Combined with text, data, and visual analysis, more accurate text can be generated to meet the filing requirements of judicial documents.

5. Conclusion

In this paper, we put forward the importance of text generation technology in the intellectualization of judicial system and then summarize the current text generation techniques according to text input, data input, and visual information input. After that, we propose how to apply these techniques to the actual judicial text generation. Particularly, the intelligent dialog system and text summary technology can be employed to intelligent consultation and intelligent extraction in smart justice. Moreover, data-driven text generation can be used to automatically generate judicial reports. The generation of image, video, and text can meet the requirements of judicial document filing. Finally, we discuss the text generation techniques according to the characteristics of judicial text and judicial work and locate the problems and challenges in their application to judicial work.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] P. Br&Dotzillon, "Context in problem solving: a survey," *The Knowledge Engineering Review*, vol. 14, no. 14, pp. 47–80, 1999.

[2] B. Lavoie and O. Rainbow, "A fast and portable realizer for text generation systems," in *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Washington, DC, USA, March 1997.

[3] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, vol. 1, pp. 285–295, Hong Kong, China, May 2001.

[4] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[5] V. Tran, M. L. Nguyen, and K. Satoh, "Building legal case retrieval systems with lexical matching and summarization using a pre-trained phrase scoring model," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pp. 275–282, Montreal, QC, Canada, June 2019.

[6] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing [exploratory dsp]," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 145–154, 2010.

[7] M. R. Keyvanpour, M. Javideh, and M. R. Ebrahimi, "Detecting and investigating crime by means of data mining: a general crime matching framework," *Procedia Computer Science*, vol. 3, pp. 872–880, 2011.

[8] E. Loper and S. Bird, "Nltk: The Natural Language Toolkit," https://arxiv.org/abs/cs/0205028.

[9] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 1958.

[10] R. Barzilay and M. Elhadad, "Using lexical chains for text summarization," *Advances in Automatic Text Summarization*, pp. 111–121, MASS, Amherst, MA, USA, 1999.

[11] H. P. Edmundson, "New methods in automatic extracting," *Journal of the ACM*, vol. 16, no. 2, pp. 264–285, 1969.

[12] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," *Advances in Automatic Summarization*, pp. 55–60, MASS, Amherst, MA, USA, 1999.

[13] M. Osborne, "Using maximum entropy for sentence extraction," in *Proceedings of the ACL-02 Workshop on Automatic Summarization*, pp. 1–8, Association for Computational Linguistics, PA, USA, July 2002.

[14] M. Kågebäck, O. Mogren, N. Tahmasebi, and D. Dubhashi, "Extractive summarization using continuous vector space models," in *Proceedings of the 2nd Workshop on Continuous Vector Space Models and Their Compositionality (CVSC)*, pp. 31–39, Gothenburg, Sweden, April 2014.

[15] A. Fiori, *Trends and Applications of Text Summarization Techniques*, IGI Global, Hershey, PA, USA, 2020.

[16] Y. Dong, "A survey on neural network-based summarization methods," https://arxiv.org/abs/1804.04589.

[17] W. Yin and Y. Pei, "Optimizing sentence modeling and selection for document summarization," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina, July 2015.

[18] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," https://arxiv.org/abs/1603.07252.

[19] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: a recurrent neural network based sequence model for extractive summarization of documents," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, February 2017.

[20] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 93–98, CA, USA, June 2016.

[21] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, and G. Cagler, "Abstractive text summarization using sequence-to-sequence rnns and beyond," https://arxiv.org/abs/1602.06023.

[22] A. See, P. J. Liu, and C. D. Manning, "Get to the point: summarization with pointer-generator networks," https://arxiv.org/abs/1704.04368.

[23] Q. Zhang, C. Bai, L. T. Yang, Z. Chen, P. Li, and H. Yu, "A unified smart Chinese medicine framework for healthcare and medical services," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, 2021.

[24] Q. Zhang, C. Bai, Z. Chen et al., "Deep learning models for diagnosing spleen and stomach diseases in smart chinese medicine with cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 4, Article ID e5252, 2019.

[25] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "Incremental deep computation model for wireless big data feature learning," *IEEE Transactions on Big Data*, vol. 6, no. 2, 2020.

[26] R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, "On the evaluation of dialogue systems with next utterance classification," https://arxiv.org/abs/1605.05414.

[27] S. S. Mohamad, N. Salim, and M. N. Jambli, "Service chatbots: a systematic review," *Expert Systems with Applications*, vol. 184, Article ID 115461, 2021.

[28] D. Ferrucci, E. Brown, J. C. Carroll et al., "Building watson: an overview of the deepqa project," *AI Magazine*, vol. 31, no. 3, pp. 59–79, 2010.

[29] A. Kalyanpur, S. Patwardhan, B. Boguraev, A. Lally, and J. C. Carroll, "Fact-based question decomposition in deepqa," *IBM Journal of Research and Development*, vol. 56, no. 3.4, pp. 13–21, 2012.

[30] A. Kalyanpur, B. K. Boguraev, S. Patwardhan et al., "Structured data and inference in deepqa," *IBM Journal of Research and Development*, vol. 56, no. 3.4, pp. 10:1–10:14, 2012.

[31] O. Vinyals and Q. Le, "A neural conversational model," https://arxiv.org/abs/1506.05869.

[32] A. Sordoni, M. Galley, M. Auli et al., "A neural network approach to context-sensitive generation of conversational responses," https://arxiv.org/abs/1506.0671.

[33] A. Kumar, O. Irsoy, P. Ondruska et al., "Ask me anything: dynamic memory networks for natural language processing," in *Proceedings of the International Conference on Machine Learning*, pp. 1378–1387, NY, USA, June 2016.

[34] Y. Zhang, Z. Gan, and L. Carin, "Generating text via adversarial training," in *Proceedings of the NIPS workshop on Adversarial Training*, vol. 21, Barcelona, Spain, December 2016.

[35] I. Goodfellow, J. A. Pouget, M. Mirza et al., "Generative adversarial nets," *Advances in Neural Information Processing Systems*, pp. 2672–2680, MIT Press, Cambridge, MA, USA, 2014.

[36] M. Jang, "Sentence transition matrix: an efficient approach that preserves sentence semantics," *Computer Speech & Language*, vol. 71, Article ID 101266, 2021.

[37] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," https://arxiv.org/abs/1701.06547.

[38] S. Gao, X. Chen, P. Li et al., "Abstractive text summarization by incorporating reader comments," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6399–6406, Honolulu, HI, USA, February 2019.

[39] C. Zhang, C. Xiong, and L. Wang, "A research on generative adversarial networks applied to text generation," in *Proceedings of the 2019 14th International Conference on Computer Science & Education (ICCSE)*, pp. 913–917, IEEE, Toronto, ON, Canada, August 2019.

[40] Y. C. Chen and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," https://arxiv.org/abs/1805.11080.

[41] Z. Li, J. Kiseleva, and M. D. Rijke, "Dialogue generation: from imitation learning to inverse reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6722–6729, Honolulu, HI, USA, February 2019.

[42] P. Kouris, G. Alexandridis, and A. Stafylopatis, "Abstractive text summarization based on deep learning and semantic content generalization," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5082–5092, Florence, Italy, July 2019.

[43] L. Song, Y. Zhang, Z. Wang, and D. Gildea, "A graph-to-sequence model for amr-to-text generation," https://arxiv.org/abs/1805.02473.

[44] L. Banarescu, C. Bonial, S. Cai et al., "Abstract meaning representation for sembanking," in *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pp. 178–186, Sofia, Bulgaria, August 2013.

[45] A. R. Fabbri, I. Li, T. She, S. Li, and D. R. Radev, "Multi-news: a large-scale multi-document summarization dataset and abstractive hierarchical model," https://arxiv.org/abs/1906.01749.

[46] J. G. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," *SIGIR*, vol. 98, pp. 335-336, 1998.

[47] E. Reiter and R. Dale, *Building Natural Language Generation Systems*, Cambridge University Press, Cambridge, 2000.

[48] S. Wiseman, S. M. Shieber, and A. M. Rush, "Challenges in data-to-document generation," https://arxiv.org/abs/1707.08052.

[49] F. O. Asahiah, "Comparison of rule-based and data-driven approaches for syllabification of simple syllable languages and the effect of orthography," *Computer Speech & Language*, vol. 70, Article ID 101233, 2021.

[50] C. Hallett, R. Power, and D. Scott, *Summarisation and Visualisation of E-Health Data Repositories*, UK E-Science All-Hands Meeting, Nottingham, UK.

[51] A. Gatt, F. Portet, E. Reiter et al., "From data to text in the neonatal intensive care unit: using nlg technology for decision support and information management," *Ai Communications*, vol. 22, no. 3, pp. 153–186, 2009.

[52] H. Banaee, M. U. Ahmed, and A. Loutfi, "Towards nlg for physiological data monitoring with body area networks," in *Proceedings of the 14th European Workshop on Natural Language Generation*, pp. 193–197, Sofia, Bulgaria, August 2013.

[53] A. S. Ramos, A. J. Bugarin, S. Barro, and J. Taboada, "Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 1, pp. 44–57, 2014.

[54] D. Gkatzia, O. Lemon, and V. Rieser, "Natural language generation enhances human decision-making with uncertain information," https://arxiv.org/abs/1606.03254.

[55] P. Liang, M. I. Jordan, and D. Klein, "Learning semantic correspondences with less supervision," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 91–99, Association for Computational Linguistics, Suntec, Singapore, August 2009.

[56] G. Angeli, P. Liang, and D. Klein, "A simple domain-independent probabilistic approach to generation," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 502–512, Association for Computational Linguistics, MA, USA, October 2010.

[57] P. K. V. Sowdaboina, S. Chakraborti, and S. Sripada, "Learning to summarize time series data," in *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 515–528, Springer, Kathmandu, Nepal, April 2014.

[58] D. Gkatzia, H. Hastie, and O. Lemon, "Comparing multi-label classification with reinforcement learning for summarisation of time-series data," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 1231–1240, MD, USA, June 2014.

[59] H. Mei, M. Bansal, and M. R. Walter, "What to talk about and how? selective generation using lstms with coarse-to-fine alignment," https://arxiv.org/abs/1509.00838.

[60] R. Lebret, D. Grangier, and M. Auli, "Neural text generation from structured data with application to the biography domain,".

[61] A. Liu, N. Xu, H. Zhang, W. Nie, Y. Su, and Y. Zhang, "Multi-level policy and reward reinforcement learning for image captioning," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, pp. 821–827, Stockholm, Sweden, July 2018.

[62] R. Puduppully, L. Dong, and M. Lapata, "Data-to-text generation with content selection and planning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6908–6915, Honolulu, HI, USA, February 2019.

[63] A. Farhadi, M. Hejrati, M. A. Sadeghi et al., "Every picture tells a story: generating sentences from images," in *Proceedings of the European Conference on Computer Vision*, pp. 15–29, Springer, Crete, Greece, September 2010.

[64] J. R. Curran, S. Clark, and J. Bos, "Linguistically motivated large-scale nlp with c&c and boxer," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pp. 33–36, Association for Computational Linguistics, Prague Czech Republic, June 2007.

[65] J. Liu, M. Yang, C. Li, and R. Xu, "Improving cross-modal image-text retrieval with teacher-student learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 3242–3253, 2021.

[66] P. Kuznetsova, V. Ordonez, T. L. Berg, and Y. Choi, "Treetalk: composition and compression of trees for image descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 351–362, 2014.

[67] Y. Yang, C. L. Teo, H. Daumé III., and Y. Aloimonos, "Corpus-guided sentence generation of natural images," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 444–454, Association for Computational Linguistics, Edinburgh United Kingdom, July 2011.

[68] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn," https://arxiv.org/abs/1412.6632.

[69] H. Fang, S. Gupta, F. Iandola et al., "From captions to visual concepts and back," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1473–1482, Boston, MA, USA, June 2015.

[70] K. Xu, J. Ba, R. Kiros et al., "Show, attend and tell: neural image caption generation with visual attention," in *Proceedings of the International Conference on Machine Learning*, pp. 2048–2057, Atlanta GA USA, June 2015.

[71] C. Zhou, J. Bai, J. Song et al., "Atrank: An attention-based user behavior modeling framework for recommendation," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, LA, USA, February 2018.

[72] L. Zhang, F. Sung, F. Liu et al., "Actor-critic sequence training for image captioning," https://arxiv.org/abs/1706.09601.

[73] Z. Ren, X. Wang, N. Zhang, X. Lv, and L. J. Li, "Deep reinforcement learning-based image captioning with embedding reward," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 290–298, HI, USA, July 2017.

[74] A. Nguyen, Q. D. Tran, T.-T. Do, I. Reid, D. G. Caldwell, and N. G. Tsagarakis, "Object captioning and retrieval with natural language," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, Seoul, Korea (South), October 2019.

[75] K. Barnard, P. Duygulu, and D. Forsyth, "Clustering art," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR*, vol. Vol. 2, IEEE, HI, USA, December 2001.

[76] H. Wang, A. Kläser, C. Schmid, and L. Cheng, "Action recognition by dense trajectories," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR*, Colorado Springs, CO, USA, June 2011.

[77] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3551–3558, Sydney, Australia, December 2013.

[78] Z. Xu, Y. Yang, and A. G. Hauptmann, "A discriminative cnn video representation for event detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1798–1807, MA, USA, June 2015.

[79] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving Deeper Into Convolutional Networks For Learning Video Representations," https://arxiv.org/abs/1511.06432.

[80] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang, "Hierarchical recurrent neural encoder for video representation with application to captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1029–1038, Las Vegas, Nevada, USA, June 2016.

[81] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, "Video paragraph captioning using hierarchical recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4584–4593, Las Vegas, Nevada, USA, June 2016.

[82] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: a large video description dataset for bridging video and language," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5288–5296, Las Vegas, Nevada, USA, June 2016.

[83] C. Xiao, H. Zhong, Z. Guo et al., "Cail2018: a large-scale legal dataset for judgment prediction," https://arxiv.org/abs/1807.02478.

[84] L. Zhong, Z. Zhong, Z. Zhao, S. Wang, K. D. Ashley, and M. Grabmair, "Automatic summarization of legal decisions using iterative masking of predictive sentences," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pp. 163–172, Montreal, QC, Canada, June 2019.

[85] G. Governatori, A. Rotolo, R. Riveret, and S. Villata, "Modelling dialogues for optimal legislation," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pp. 229–233, Montreal, QC, Canada, June 2019.

[86] Y. Kang, D. Peng, Z. Chen, and C. Liu, "Ed-gan: Judicial document generating model based on improved generative adversarial networks," *Journal of Chinese Computer Systems*, vol. 40, no. 5, pp. 1020–1025, 2019.

[87] M.-F. Moens, E. Boiy, R. M. Palau, and C. Reed, "Automatic detection of arguments in legal texts," in *Proceedings of the 11th International Conference on Artificial Intelligence and Law*, pp. 225–230, ACM, CA, USA, June 2007.

[88] N. Martínez Melis and A. Hurtado Albir, "Assessment in translation studies: research needs, Meta," *journal des traducteurs/Meta: Translators' Journal*, vol. 46, no. 2, pp. 272–287, 2001.

[89] J. Keppens, "Explainable bayesian network query results via natural language generation systems," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pp. 42–51, Montreal, QC, Canada, June 2019.