2023 4th International Conference on Computer, Big Data and Artificial Intelligence (ICCBD+AI) | 979-8-3503-7323-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCBD-AI62252.2023.00015

A Method for Extracting Information from Long Documents that Combines Large Language Models with Natural Language Understanding Techniques

1st Linjie Chen*
China Mobile Information Technology Company Limited
Shenzhen, China
chenlinjie@chinamobile.com

2nd Min Sun
China Mobile Information Technology Company Limited
Shenzhen, China
sunmin@chinamobile.com

3rd Jiarong Liu
China Mobile Information Technology Company Limited
Shenzhen, China
liujiarong@chinamobile.com

4th Pengyong Ding
China Mobile Information Technology Company Limited
Shenzhen, China
dingpengyong@chinamobile.com

5th Yuliang Ma
China Mobile Information Technology Company Limited
Shenzhen, China
mayuliang@chinamobile.com

6th Li Li
China Mobile Information Technology Company Limited
Shenzhen, China
liliit@chinamobile.com

7th Lili Qian
China Mobile Information Technology Company Limited
Beijing, China
qianlili@chinamobile.com

8th Yang Yang
China Mobile Information Technology Company Limited
Beijing, China
yangyanggh@chinamobile.com

***Abstract*—Information extraction is a very important task in natural language processing and is widely used in various industries, but due to the ever-changing types of documents, there are still challenges in terms of extraction effectiveness in practical applications. In previous research, traditional AI models were commonly used to address the issue of information extraction from long documents. However, due to challenges related to input length and the capability to understand the semantics of extended texts, the results have consistently fallen short of expectations and have not been practically implemented in industrial production.**

In this paper, we propose a solution for extracting information from long document based on large language model. While possessing the ability of long-text semantic understanding of large language models, it also effectively alleviates the hallucinations of large language models and the problem of uncontrollable output results through discriminative methods. Compared with other general methods, our approach has achieved significant performance improvement on the long-text dataset we collected. Additionally, our approach is reproducible and can be easily and quickly customized and adapted to other scenarios.

***Index Terms*—Information Extraction; Transformers; Large language models; AIGC; Natural language processing; Machine learning; Artificial intelligence;**

I. Introduction

In this chapter, we will provide a background introduction on the task of information extraction and its applications. We will also discuss the current issues and challenges in some methods based on large language models (LLMs). Finally, we will state our research objective, propose and evaluate effective methods to enhance the practical performance of information extraction.

A. Information Extraction

Information extraction is a technique in natural language processing that involves extracting structured information from unstructured or semi-structured text data. Information extraction is a pivotal task in the realm of natural language processing. It plays a crucial role in deciphering and retrieving meaningful data from vast amounts of unstructured text. This technique is extensively employed across various industries, ranging from healthcare to finance, aiding in tasks such as data mining, knowledge representation, and decision-making. Information extraction can be classified into the following types: Named Entity Recognition (NER): This is a technique for identifying and extracting specific types of entities (such as person, location, organization, etc.) from text. Relation Extraction: Extracting the relationships between entities from text. Event Extraction: Extracting events and their related entities, time, location, and other information from text.

One of the widely used techniques for information extraction is based on machine learning models, particularly deep learning models such as recurrent neural networks (RNNs) [1], and transformer models like BERT [2]. These models are trained on large datasets to learn the patterns and contextual meaning of words and sentences, which helps in accurate extraction of information.

There are widely application scrnarios on information extration, such as extracting invoice numbers and other fields in invoice documents, extracting key elements such as company names and contract amounts in contract documents, and ex-

979-8-3503-7323-3/23/$31.00 ©2023 IEEE
DOI 10.1109/ICCBD-AI62252.2023.00015

tracting structured data from a large number of unstructured domain documents to form industry-related knowledge graphs.

B. Large Language Models

Over the past two decades, language modeling methods have been widely used for language understanding and generation, including statistical methods and neural language models. In recent years, researchers have begun to use large-scale corpora to pre-train models based on the Transformer architecture [3], and it has been found that pre-trained language models exhibit strong abilities in processing and solving various natural language processing tasks. Moreover, when the parameter size exceeds a certain level, this larger language model achieves significant performance improvements and exhibits capabilities that are not present in smaller models, such as context learning.

Since the release of OpenAI's ChatGPT [4] in 2022, the research on large language models has experienced explosive development in both academic and industrial fields, and has attracted widespread attention worldwide. Large model technology has had a significant impact on the entire academic and even social development, and has completely changed the way people develop and use AI algorithms.

C. Research Objective

If we directly utilize the zero-shot inference capability of LLMs for information extraction tasks, we often can only extract relatively simple entities, such as names, addresses within a short sentence. However, for more complex extraction fields or entities specific to certain scenarios, the performance often falls short of expectations. Additionally, the outputs of large models are uncertain, and in industrial applications, measures are needed to impose constraints.

Although researchers have already proposed several methods to address this issue currently, such as LoRA fine-tuning [5] and In-context Learning [6]. In information extraction tasks, fine-tuning or in-context learning using the aforementioned approaches indeed result in some performance improvements. However, there are still bottlenecks in extracting information from long document. One issue is data annotation; manually annotating long texts is challenging, time-consuming, and prone to errors. On the other hand, the benefits of in-context learning or few-shot learning are not evident in long-text scenarios, and the model's output might still be influenced by hallucinations.

Therefore, we believe it is necessary to find a method that can enhance extraction performance on long document extraction tasks and ensure that the inference result meets the requirements.

In summary, the main contributions of this paper are twofold: first, it proposes a comprehensive solution for long document information extraction based on large language models; second, it offers a traditional model-based approach to address the hallucination and instability issues associated with the outputs of large models.

II. Preliminaries

NLP can be divided into two major tasks: natural language understanding (NLU) and natural language generation (NLG). NLU focuses on parsing and comprehending language content, while NLG concentrates on creating and producing coherent text. We have divided the research related to information extraction into two categories: NLU-based methods and NLG-based methods.

A. NLU-based methods

There are several NLU-based methods for information extraction, including token classification and Conditional Random Fields (CRF). CRFs are a type of probabilistic graphical model that can be used for labeling and segmenting sequential data, they are particularly useful for tasks where context is essential. For instance, in NER, the word ”Apple” might be a ”FRUIT” in one context and a ”COMPANY” in another. The Conditional Random Fields (CRF) formula for a linear chain CRF is given by:

$$P(y|x) = \frac{\exp\left(\sum_i \lambda_i f_i(y,x) + \sum_j \mu_j g_j(y,x)\right)}{\sum_{y'} \exp\left(\sum_i \lambda_i f_i(y',x) + \sum_j \mu_j g_j(y',x)\right)} \quad (1)$$

This formula calculates the probability of a label sequence y given an input sequence x. The numerator represents the weighted sum of the feature functions for the given label sequence, and the denominator normalizes over all possible label sequences y'.

Token classification is a common method used in Named Entity Recognition, each token or a word in a sentence is classified into predefined categories or labels. For example, in the sentence ”Barack Obama was born in Hawaii”, ”Barack Obama” might be classified as a 'PERSON' and ”Hawaii” as a ”LOCATION”.

The objective of token classification can be represented by:

$$y_t = f(x_t; \theta) \quad (2)$$

Where:

- y_t is the output label at time t.
- x_t is the input element at time t.
- f is a parameterized function used to map from input elements to output labels.
- θ represents the parameters of the model.

However, these methods have some shortages. For instance, small models (parameter size) often lack better semantic understanding, and the context is typically limited. For example, the length of BERT's pre-trained corpus is usually only 512, requiring truncation for longer sentences. On the other hand, the cost of customizing for specific scenarios is high. It typically requires hundreds or even more annotated data. Moreover, these scenarios tend to be relatively closed-off, making it challenging for models to be effectively transferred and reused.

B. NLG-based methods

Before the emergence of the ChatGPT series of LLMs, Lu and others propose a method Univeral Information Extraction [7], using the T5 generative model as the backbone, proposed information extraction through prompting.

$$H = \text{Encoder}(s_1, \ldots, s_{|s|}, x_1, \ldots, x_{|x|}) \tag{3}$$

$$y_i, h_{d_i} = \text{Decoder}\left(\left[H; h_{d_1}, \ldots, h_{d_{i-1}}\right]\right) \tag{4}$$

This method designed a unified prompt template for various types of information extraction tasks, allowing for easy training adaptation across different scenarios. However, this approach does not possess zero-shot capabilities, and its performance in few-shot learning is not satisfactory.

Information extraction based on LLMs is more convenient. Sometimes, there's no need for training step. By providing the LLMs with text input and prompts that align with natural language expression logic, it can accomplish the task. With the development of LLMs, there have also been a number of applications based on large models, such as LMDX [8], which uses large models for document information extraction and localization, and InstructIE [9], which proposes an information extraction method based on instructions and large models and constructs a corresponding dataset. However, their primary focus is on the improvement of extracting key information from short texts or data from single-page receipts, documents etc. They haven't extensively optimized for long document scenarios.

III. Methods

A. Problem Definition

Unlike extracting information from a single sentence, extracting information from an extremely long text requires relying on more context and semantic analysis capabilities. If the input document is in the format of a PDF or image, the text will be recognized through OCR (Optical Character Recognition). The task of extracting key information from long documents can be defined as following:

1) Input:

- A long document D where $D = \{d_1, d_2, \ldots, d_n\}$ and d_i represents the i-th word or token in the document.
- A set of field names F where $F = \{f_1, f_2, \ldots, f_m\}$ that need to be extracted from the document.

2) Task:

- For each field name f_j in F, identify a corresponding segment S_j in D such that S_j contains the information related to f_j.

3) Output:

- A set of extracted segments $\mathcal{S}$ where $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ corresponding to each field name in F.

Our goal is to design a reasonable process that, under the premise of low development costs, can effectively extract key information from long documents. In the following content, we will provide a detailed introduction to our technical solution, which is the cascading approach of a large model (Transformer-Decoder based) followed by a smaller model (Transformer-Encoder based). This allows us to leverage the extensive contextual understanding capabilities of the large model while avoiding output uncertainties.

TABLE I
Prompt Template 1: Single Key Extraction

Please extract the information corresponding to [key] from the contract below based on the hints provided. When returning the results, use the JSON format, containing a key-value pair with the key being {key}. Hint: The description of [key] in the contract includes {key_words}. Contract content: {ocr_result}.

TABLE II
Prompt Template 2: All Keys Extraction

Please extract the key information corresponding to each item in the keyword list from the contract information below based on the hints provided. The keyword list I specified is enclosed in [] symbols. You need to make a comprehensive judgment based on the context and semantics to extract accurate key information. When returning the results, use the JSON format, containing multiple key-value pairs, where the key is the keyword I specified, and the value is the extracted result. Please only output the results in JSON format and do not include any other extraneous text. Hint: Synonyms for [key] include {key_words} Contract content: {ocr_result}.

B. Prompt Template Design

We have designed two types of prompts. The first prompt template shown in table I extracts only a single field from a long document at a time, so if there are k fields, it requires k large language model inferences. The second prompt template shown in table II extracts all field information from the long document in one go. The issue with this prompt is that if there are many fields to extract, the effectiveness and accuracy might be compromised.

C. Output Formatter

Although we have tried to include constraints on the desired return format (for example, in the template above, we hope to return key-value pairs in JSON format) in the prompt, we found in actual testing that even when using the SOTA LLMs such as ChatGLM2-6B [1] and Baichuan2-13B-Chat [10], through zero-shot or few-shot learning, the output of the large model often does not conform to our expected format. Therefore, we refer to the inference output of the large model as an intermediate result.

To obtain the final result that meets the format requirements from the intermediate result, one approach is to introduce rules for post-processing to obtain the JSON format result. The limitation of this approach is that it cannot exhaust all

[1] https://github.com/THUDM/ChatGLM2-6B

possibilities, and relying on rules cannot satisfy all scenarios. Another approach is to annotate more data for fine-tuning. However, this method relies heavily on a large amount of custom annotated data and has poor transferability.

Therefore, we introduced a NLU-based model in the post-processing stage. UIE[2] is a model pre-trained using ERNIE as the backbone, with a large amount of information extraction corpus. It possesses powerful zero-shot inference capabilities. UIE adopts a start & end position pointers decoding method, which can effectively transform unstructured text into structured data output. The decoding methods of UIE can be represented by the following equations:

1) Computing logits:

$$\text{logits} = W \times \text{sequence\_output} + b \tag{5}$$

Where W is the weight matrix and b is the bias.

2) Computing probabilities using the sigmoid function:

$$\text{prob} = \frac{1}{1 + e^{-\text{logits}}} \tag{6}$$

In details, Sequence output from ERNIE is passed through two linear layers to get start and end position logits respectively. These logits are then squeezed to remove the last dimension, and passed through a sigmoid function to get start and end position probabilities.

D. Workflow

The overall workflow of our method is shown in Figure. 1, it's equivalent to performing a secondary extraction on the long document. First, with the help of the LLM's long-text understanding capabilities, we obtain an intermediate result. However, due to the uncertainty of the output from large language models, intermediate results still need post-processing to be converted into a fixed key-value JSON format result. Thus, the intermediate result is organized using UIE.

UIE possesses a robust capability for few-shot learning and is one of the frequently used models in the information extraction domain. The input for UIE is plain text, and by merely annotating a small amount of data in the domain-specific text for fine-tuning, the extraction performance sees a significant improvement.

IV. Experiments

A. Dataset

We have prepared a dataset for comparison and evaluation. It consists of 70 Chinese contract samples collected from public websites, with an average text length of 4083 characters (excluding attachments). We manually annotated 8 fields in the contracts, including the contract signing date, the buyer's company, the seller's company, etc. We divided 70 documents into 10 training sets, 20 validation sets, and 40 test sets.

In subsequent experiments, 10 training set samples and 2 0 validation set samples were primarily used for training the UIE, T5, and Bert models. Given the extended length of the samples, we filtered out sentences containing key field information from the samples for annotation and training. This ensures that the length of the training data meets the requirements of the model's max_length parameter.

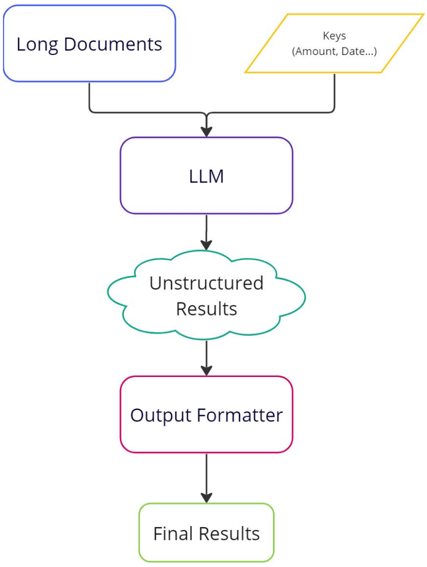

Fig. 1. The overall workflow of our method. Input long documents along with the keys to be extracted. The LLM produce an intermediate coarse-grained results. This output is then fed into the output formatter and get the final results in a fixed format.

B. Experiment Setup

We chose Baichuan2-13B-Chat and ChatGLM2-6B as our LLM backbones, with parameter sizes of approximately 13 billion and 6 billion, respectively. As for UIE, we selected UIE-base, which has an estimated parameter count of about 110 million. It consists of 12 hidden layers, a hidden size of 768, and a vocabulary size of 40,000.

The experiment uses F1 as the primary evaluation metric. It is the harmonic mean of the precision and recall, where precision is the number of correct positive results divided by the number of all positive results, and recall is the number of correct positive results divided by the number of positive results that should have been returned.

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{7}$$

We set up two main sets of experiments. Experiment one compares the F1 scores of extracting one field at a time versus extracting all fields at once. Experiment two compares the results of other methods and the method of LLM + UIE. The experiment device is one NVIDIA A100 GPU with 40G of memory.

C. Performance

1) Experiment 1: Experiment one primarily evaluates the performance of extracting a single field and extracting all

[2] https://github.com/PaddlePaddle/PaddleNLP/tree/develop/model_zoo/uie

fields at once. Table III presents the results of Experiment 1. Based on the experimental results, the F1 score is highest when using a single key combined with Baichuan2-13B-Chat. Additionally, from the output of the LLMs, we identified two main problems. First, when we combine all fields into a single prompt for key information extraction, the large model is more prone to hallucinations and repetitions. Second, whether extracting a single field or extracting all fields at once, the LLM's output cannot guarantee that the returned result is a directly parsable JSON string.

TABLE III
EXPERIMENT 1 RESULTS

Model Configuration	**F1**
ChatGLM2-6B + UIE ◇	82.86
Baichuan2-13B-Chat + UIE ◇	**89.98**
ChatGLM2-6B + UIE △	49.62
Baichuan2-13B-Chat + UIE △	56.17

◇ represents extracting a single field in an inference, while △ represents extracting all fields in an inference.

2) Experiment 2: Table IV presents the results of Experiment 2. In traditional methods, using UIE alone significantly outperforms the results of training Bert and T5 from scratch, indicating that pre-training on large-scale information extraction corpora can notably enhance the F1 score. The inclusion of LLM indeed brings a more pronounced improvement in extraction performance, which can be attributed to their superior ability to understand longer contexts.

In the experiments, the combination of single-key extraction + Baichuan-13B-Chat + UIE achieved the highest F1 score on the test set. Additionally, post-processing with the UIE model, as opposed to direct processing using regular expressions, significantly improved the final extraction results, thanks to UIE's robust capabilities in sentence-level information extraction.

TABLE IV
EXPERIMENT 2 RESULTS

Model Configuration	**P**	**R**	**F1**
Bert(From scratch)	11.34	17.06	13.62
T5(From scratch)	19.75	15.68	17.48
UIE	36.71	41.54	38.98
ChatGLM2-6B + Regular Rules	61.21	68.39	64.60
ChatGLM2-6B + UIE	80.52	78.93	79.72
Baichuan2-13B-Chat + Regular Rules	69.01	61.80	65.21
Baichuan2-13B-Chat + UIE	**88.57**	**91.43**	**89.98**

P represents precision, and R represents recall.

V. CONCLUSION

This paper proposes a solution that combines a LLM with a natural language understanding sub-model to extract key information from long documents. From the experimental results, it's evident that the introduction of the LLM significantly improves extraction performance. The traditional Transformer Encoder model further aids in post-processing the output of the large model to enhance extraction results. It is an innovative approach based on mainstream LLM and can be quickly implemented in various real-world scenarios.

However, the approach has its limitations. It primarily revolves around assisting and improving the outputs of the large model, requiring a two-stage process. Future work could explore further refinements during the fine-tuning phase of the large model, investigating if the long document extraction problem can be addressed in a single inference step.

REFERENCES

[1] Z. Lu, V. Sindhwani, and T. N. Sainath, *Learning compact recurrent neural networks* ICASSP. IEEE, 2016, pp. 5960–5964.

[2] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* NAACL-HLT 2019: 4171-4186

[3] Ashish Vaswani et al. *Attention is All you Need.* NIPS 2017: 5998-6008

[4] Long Ouyang et al. *Training language models to follow instructions with human feedback.* NeurIPS 2022

[5] Edward J. Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models* ICLR 2022

[6] Yinheng Li. *A Practical Survey on Zero-shot Prompt Design for In-context Learning.* CoRR abs/2309.13205: (2023)

[7] Yaojie Lu et al. *Unified Structure Generation for Universal Information Extraction.* ACL (1) 2022: 5755-5772

[8] Vincent Perot et al. *LMDX: Language Model-based Document Information Extraction and Localization.* CoRR abs/2309.10952: (2023)

[9] Honghao Gui et al. *InstructIE: A Chinese Instruction-based Information Extraction Dataset.* CoRR abs/2305.11527: (2023)

[10] Baichuan. *Baichuan 2: Open Large-scale Language Models* arXiv preprint arXiv:2309.10305 2023