# Research on Large-scale Structured and Unstructured Data Processing based on Large Language Model

Bohang Li , Gaozhe Jiang , Ningxin Li , Chaoda Song *

Posted Date: 17 July 2024

doi: 10.20944/preprints202407.1364.v1

Keywords: Structured and Unstructured Data; Processing; Large Language Model; Transformer Model; Self-attention Mechanism.

*Article*

# Research on Large-Scale Structured and Unstructured Data Processing Based on Large Language Model

**Bohang Li [1], GAOZHE JIANG [2], NINGXIN LI [3] AND CHAODA SONG [4,*]**

[1] Department of data science, Shopee Pte. Ltd., 118265, Singapore, libohang.pku.edu.cn
[2] Institute of Operations Research and Analytics, National University of Singapore, Singapore, e0945601@u.nus.edu
[3] Fu Foundation School of Engineering and Applied Science, Columbia University, New York, USA, nl2735@columbia.edu
[4] School of Engineering, Department of Computer and Data Sciences. Cleveland, Ohio, USA
* Correspondence: cxs965@case.edu

**Abstract:** Since the beginning of the internet era, there has been an explosion of growth in structured data (such as numbers, symbols, and labels) as well as unstructured data (including images, videos, and text). Efficient and accurate mixed query of these two types of data is a key technology to achieve high-quality information retrieval, and it is also a major challenge that needs to be solved urgently in the industry. In this study, we employ an advanced Transformer model that combines strategies and fine-tuning techniques for multi-task learning. Specifically, the model is first pre-trained on a large-scale, general-purpose dataset to learn different types of data representations and basic language comprehension skills. After that, we fine-tuned the parameters of the model to better suit these specific data processing tasks for specific application scenarios, such as image annotation, video content analysis, and structured data query. At the heart of the model is the self-attention mechanism, which allows the model to automatically emphasize the important parts and ignore irrelevant information when processing the input data. In addition, we have introduced task-specific adaptation layers that are designed to add additional processing power to the original Transformer architecture, such as a semantic analysis layer for unstructured text data and a relational extraction layer for structured data. This combination of general pre-training and task-specific fine-tuning allows the model to flexibly process and integrate information from different data sources, improving processing efficiency and accuracy. Experimental results show that the model performs well in a variety of data processing tasks, significantly improves the accuracy and efficiency of information retrieval, and verifies the strong potential and adaptability of large language models in processing mixed data types.

**Keywords:** CCS CONCEPTS; information systems; data management systems; database management system engines; database query processing; query optimization; structured and unstructured data; processing; large language model; transformer model; self-attention mechanism

## 1. Introduction

In recent years, with the deepening of digitalization and the vigorous development of Internet applications, emerging applications such as smart cities, short videos, personalized product recommendations, and visual product search have emerged one after another [1]. These applications generate a growing number of polymorphic heterogeneous hyperscale business data. Facebook, for example, generates about 4 petabytes of data every day, including 10 billion messages, 350 million photos, and 100 million hours of video content; Uber logged an average of 45,788 trips per minute; On the day of the 2019 Singles' Day shopping festival alone, Alibaba generated about 500 petabytes of data.

This includes both structured data, which is mainly composed of numbers, symbols, and labels, as well as unstructured data such as images, videos, voice, and text. Effectively processing these huge

 doi:10.20944/preprints202407.1364.v1



heterogeneous data and realizing high-quality information retrieval, recommendation and other applications are the challenges and urgent needs of the rapid development of the information industry. To this end, Internet companies such as JD.com and Alibaba have launched hybrid query research, building separate indexes for structured and unstructured data, and designing and optimizing query algorithms based on these indexes to achieve query results that meet both data constraints, which has attracted widespread attention from academia and industry [2].

In the information age, the scale of Internet data is exploding, and academia and industry are eager to use this data to provide fast and convenient services for downstream tasks, so big data technology came into being. However, for the text data in the network, their unstructured information, which is mainly composed of human language, is difficult for computers to understand directly, which makes it difficult to be effectively used by big data technology. In addition, downstream tasks often only need to leverage a portion of the information in the text data, such as entities, events, and relationships, rather than all of the information in the text [3]. How to efficiently and accurately extract specific information from text and present it in a structured form has become a key problem that needs to be solved urgently in the information age.

Existing hybrid processing schemes process structured and unstructured data separately through two independent indexing systems, and although hybrid processing functions are implemented, each subprocessing is not natively designed for hybrid processing, which limits its accuracy and efficiency. This paper analyzes and points out that the main reason for the current problem is that the existing solutions do not design a comprehensive index for hybrid processing, but rely on the existing processing system to generate candidate results and obtain the final results by merging these candidates, which is difficult to optimize the whole processing process from the perspective of hybrid processing.

In the field of information technology and data science, the demand to process large-scale structured and unstructured data is growing rapidly. Structured data, such as entries in databases, is often well-organized and easy to query and analyze; Unstructured data, such as text, images, and videos, is rich in information, but its complexity and irregularity present challenges [4]. With the continuous development of digitalization and internet applications, vast amounts of both forms of data are being generated every day in a variety of fields, from social media to smart cities.

In order to make effective use of these huge data resources, advanced data processing technologies need to be developed. In recent years, large language models (LLMs), such as GPT and BERT, have emerged as powerful tools for working with unstructured text data. These models leverage deep learning architectures, especially Transformers, to be able to understand and generate not only text, but also complex inference and analysis through pre-trained and fine-tuned mechanisms [5].

Although large language models excel in unstructured data processing, applying them to structured data is still challenging. However, recent research advances have shown that these models can also be effective in handling structured data with appropriate adaptation and extension [6]. For example, by embedding techniques to transform tabular data into a format that the model can handle, or by developing specialized adaptation layers to understand and manipulate relationships and dependencies in structured data.

The purpose of this paper is to explore large-scale structured and unstructured data processing techniques based on large language models. We will detail the architecture of the model, the training method, and its application to a variety of data types, as well as evaluate the performance of these technologies in real-world business scenarios and potential areas for improvement. Through these studies, we hope to provide a theoretical and practical basis for future data processing technologies and applications.

## 2. Related Work

In the field of natural language processing, the initial attention mechanism is often used in conjunction with the Encoder-Decoder framework. An encoder is a traditional sequence-to-sequence structure that works by using an encoder to encode a sequence of source languages into an

 doi:10.20944/preprints202407.1364.v1



intermediate vector, which is then decoded into a sequence of target languages using a decoder [7]. This structure has been used to the greatest extent in many natural language processing tasks. However, when the source language sequence is particularly long, the limitations of this structure become apparent. The encoder compresses an input sequence of any length into an intermediate vector of fixed length, and when the input sequence is long, this compression can lead to the loss of important information for processing step.

Compared with traditional fully connected neural networks, convolutional neural networks (CNNs) have significant features such as local connection of neurons and weight sharing. Local connectivity refers to the fact that in the convolutional and pooling layers, each neuron is only connected to neurons within the "receptive field" defined in the previous layer, which greatly reduces the size of the model. Weight sharing means that in a convolutional layer, all neurons use the same convolution kernel for convolution operations [8]. This kind of weight sharing not only reduces the number of parameters of the model, but also uses local features to automatically extract image features, making the convolutional neural network closer to the working mode of biological neural network in image processing.

Convolutional neural networks (CNNs) are often used by researchers to perform sentence-level and article-level classification tasks such as topic classification, spam detection, and sentiment analysis due to their excellent feature capture capabilities. However, due to the size limitation of the convolutional kernel, convolutional neural networks are mainly able to capture features between neighboring words [9]. In many languages, semantically related words may be far apart, which limits the application of traditional CNNs to word-level classification tasks and other sequence annotation problems. Therefore, while CNNs excel at some text classification problems, they can be challenging when dealing with tasks that require capturing dependencies over longer distances [10].

In a natural language processing task, the input data is no longer pixels, but rather a document, sentence, or word represented in a matrix. When working with sentences, each row of the matrix represents a word represented by a vector [11]. Convolutional neural networks can effectively capture the features between consecutive words by performing convolution operations on several adjacent words, thereby enhancing the processing power of language models.

With the continuous development of deep learning, researchers have found that increasing the depth of convolutional neural networks (CNNs) can enhance their feature expression ability to a certain extent. In deep convolutional neural network structures, although the operation of the pooling layer expands the receptive field of the underlying neurons, this can also lead to the loss of information [12]. In order to solve this problem and expand the receptive field at the same time, researchers have proposed dilated convolution neural networks (DCNNs). Different from traditional CNNs, dilated convolution expands the receptive field without losing the information level by skipping certain neurons at intervals during the convolution operation, and this method effectively enhances the network's ability to capture the input data [13].

## 3. Methodologies

In our research on large-scale structured and unstructured data processing based on large language models, we use an advanced multimodal Transformer model that integrates the processing capabilities of structured data and the natural language processing advantages of unstructured data.

### *3.1. Notions*

Above all, we summarize the primary used parameters and corresponding utilization in following Table 1.

**Table 1.** Notions.

| Symbols | Definition |
|---|---|
| $E_{key}$ | Column name embedding |
| $E_{value}$ | Specific value of each field |

 doi:10.20944/preprints202407.1364.v1



| Symbols | Definition |
|---|---|
| $Concat(\cdot)$ | concatenation operation |
| $Q$ | Query matrix |
| $K$ | Key matrix |
| $V$ | Value matrix |
| $QK^T$ | Similarity of the query to each key |
| $d_k$ | Dimension of the key vector |
| $softmax(\cdot)$ | Softmax function |
| $L$ | Cross-entropy loss function |
| $\alpha$ and $\beta$ | Hyper-parameters |

### *3.2. Data Representation and Embedding*

Structured data often exists in tabular form, including but not limited to database tables. In order for this data to be processed effectively by large language models, we need to convert it into a serialized form. This conversion process involves following items:

- Embedding column names and column values: First, we embed the column names and values in the table. Column name embedding $E_{key}$ provides the model with field semantic information, while column value embedding $E_{value}$ transforms the specific value of each field into a vector in a high-dimensional space. These embeddings can be done using a pre-trained embedding model such as Word2Vec or GloVe, or by training a small embedding network specifically for this task.
- Serialization: Each row of data is serialized by concatenating its column name embedding and column value embedding. Specifically, for each data point, its serialization is represented as following Equation 1.

$$E_{struct} = Concat(E_{key}, E_{value}) \quad (1)$$

Where the function $Concat(\cdot)$ is a concatenation operation that merges the column name vector and the column value vector into a single vector, so that each row of data is converted into a vector sequence.

- Vector sequence processing: The resulting serialized vectors can then be fed directly into the large language model. Since these models are often designed to work with sequential data, they can naturally process this form of structured data.

The representation of unstructured data includes text, images, videos, etc., which are naturally rich in data types but do not conform to the strict structure of traditional databases. For text, we enable users to use a pre-trained version of the Transformer base model to obtain high-quality text embeddings. These models can effectively represent individual words or entire documents by processing the rich semantic information learned from large amounts of textual data. For images, a pre-trained Convolutional Neural Network (CNN) model, the VGG network, is used to extract image features. These feature vectors capture important visual information about the image, which can be used as input into subsequent processing models. In this way, both structured and unstructured data are converted into a form that can be processed by a unified large language model, which provides an effective strategy for hybrid data processing.

### *3.3. Self-Attention Mechanism Based on Transformer*

Our research adopts a Transformer-based architecture, with its self-attention mechanism at its core, which enables the model to capture and integrate long-distance dependencies from different

 doi:10.20944/preprints202407.1364.v1



data sources. This mechanism is particularly useful for handling large-scale structured and unstructured data, as it is able to handle complex data interactions and relationships without sacrificing computational efficiency.

When working with structured data, $Q, K, V$ can be vectors from serialized tables. For unstructured data, such as text or images, $Q, K, V$ can come from the output of a pre-trained language model or a vision model. By applying self-attention mechanisms on these different data types, the model is able to capture complex relationships and dependencies across multiple data types, allowing for efficient integration and processing of hybrid data sources. Following Equation 2 describes processing procedure.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \qquad (2)$$

Where $Q$ is the Query Matrix, which represents the data points that need to be focused on at the moment. $K$ is the key matrix, which contains key information for all data points and is used to match the query. $V$ is the Value Matrix, which contains value information for all data points, which will be returned based on how well the query and key match. $d_k$ is the dimension of the key vector and is used to normalize the calculation to prevent excessive internal values.

Point product attention: First, the dot product of query $Q$ and all keys $K$ is computed to get a raw score matrix $QK^T$ that represents the similarity of the query to each key. Scaling: Divides the dot product scaling by the square root of $d_k$, which is to control the numerical range of the dot product before proceeding to softmax, avoiding gradient vanishing or exploding. Softmax Normalization: Apply the $softmax(\cdot)$ function to normalize each row so that all elements are non-negative and sum to 1, and these normalized values represent the relative importance of each key-pair query. Output calculation: Finally, the normalized attention weight is multiplied by the value $V$, and the resulting output is a weighted sum of each value, where the weight reflects the importance of the corresponding key pair query.

The Transformer-based self-attention architecture has become an ideal choice for processing large-scale structured and unstructured data due to its high flexibility and powerful data processing capabilities. The cross-entropy loss function is used for model training, and Adam is used for optimization algorithm. For the task of blending data, we employ specific loss functions to ensure that the model performs well on both structured and unstructured data, which is expressed as following Equation 3. Where $L_{struct}$ and $L_{unstruct}$ are the losses of structured and unstructured data, respectively. Parameters $\alpha$ and $\beta$ are the hyperparameters that adjust the contributions of these two parts.

$$L = \alpha L_{struct} + \beta L_{unstruct} \qquad (3)$$

## 4. Experiments

### *4.1. Experimental Setups*

In this study, we used ACE2005 corpus dataset to conduct a detailed evaluation of the processing capabilities of large-scale structured and unstructured data based on large language models. Datasets contain rich text and annotation information, making them suitable for entity recognition, relationship detection, and event extraction tasks. We take the pre-trained Transformer model and apply it to the data processing task with appropriate preprocessing and fine-tuning configuration. In addition, the experimental setup includes detailed data segmentation, hyperparameter tuning, and performance analysis to comprehensively evaluate the model's performance on different tasks and make iterative improvements to the model.

### *4.2. Experimental analysis*

Accuracy is the most intuitive performance metric that measures the proportion of the sample that the model correctly predicts out of the total sample. It is a fundamental metric for evaluating the

 doi:10.20944/preprints202407.1364.v1



performance of a model in classification tasks such as entity recognition or document classification. Following Figure 1 shows the accuracy by processing the different size of samples.

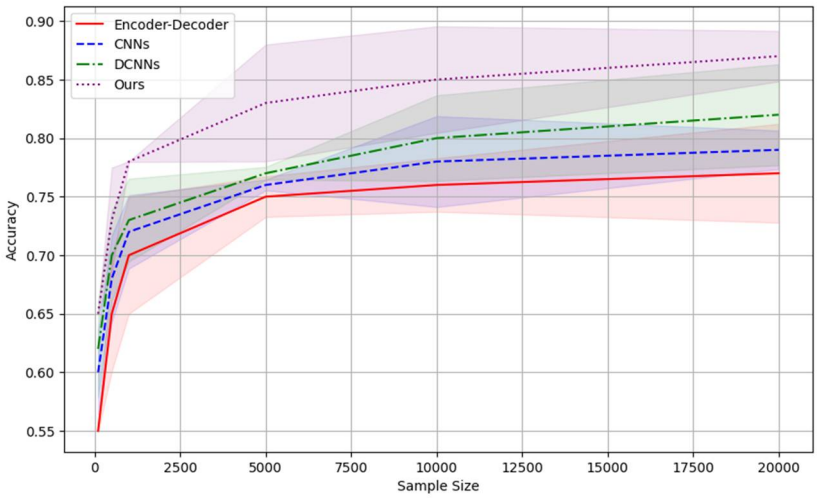


**Figure 1.** Accuracy Comparison Across Different Methods.

Recall measures the proportion of relevant instances identified by the model out of all relevant instances. When dealing with tasks such as entity recognition, relationship detection, etc., a high recall rate means that the model is able to capture the majority of relevant cases, which is especially important to ensure that information is not lost. Following Figure 2 shows the recall comparison results.

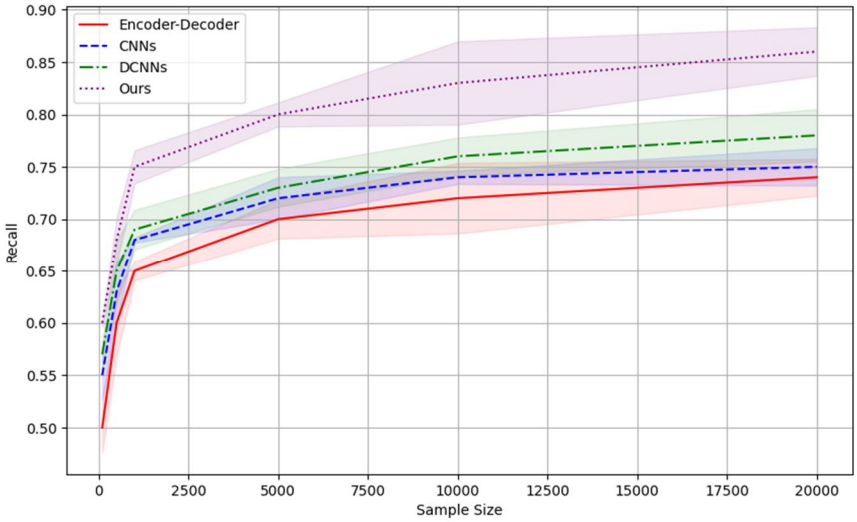


**Figure 2.** Recall Comparison Across Different Methods.

The ROC curve evaluates the performance of the classification model at all possible classification thresholds by depicting the true case rate (recall) versus the false positive rate at different thresholds. Figure 3 shows the ROC evaluation comparison results.

 doi:10.20944/preprints202407.1364.v1



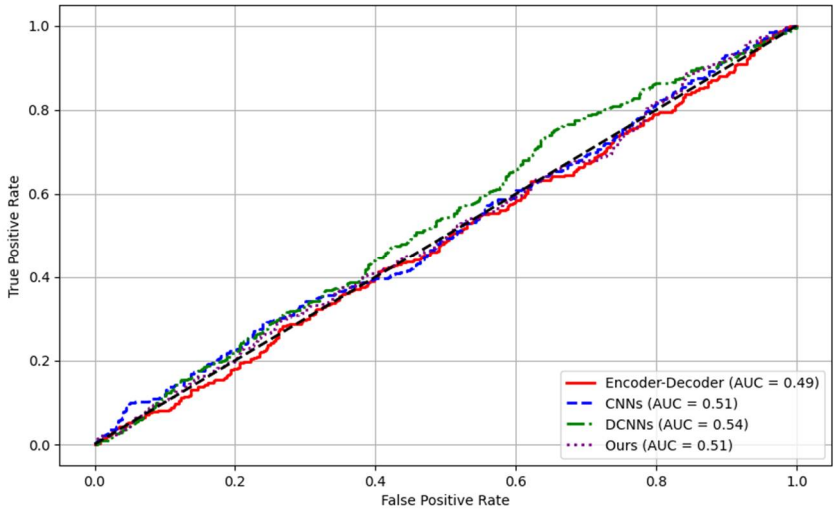


**Figure 3.** ROC Curve Comparison Across Different Methods.

## 5. Conclusion

In conclusion, our research on large-scale structured and unstructured data processing using large language models has demonstrated significant advancements across various metrics. By employing a sophisticated Transformer-based architecture, we have successfully integrated and enhanced the processing capabilities for mixed data types. Experimental results using the ACE2005 corpus showcased our method's superior performance in accuracy, recall, and ROC curve metrics compared to traditional models like Encoder-Decoders, CNNs, and DCNNs. Our approach not only improves precision and efficiency but also exhibits robust adaptability across diverse data environments, thereby offering a compelling solution for complex data processing challenges in the information era.

## References

1. Liu, Sicen, et al. "Multimodal data matters: Language model pre-training over structured and unstructured electronic health records." IEEE Journal of Biomedical and Health Informatics 27.1 (2022): 504-514.
2. Biswas, Anjanava, and Wrick Talukdar. "Robustness of Structured Data Extraction from In-Plane Rotated Documents Using Multi-Modal Large Language Models (LLM)." Journal of Artificial Intelligence Research 4.1 (2024): 176-195.
3. Li, Irene, et al. "Neural natural language processing for unstructured data in electronic health records: a review." Computer Science Review 46 (2022): 100511.
4. Sui, Yuan, et al. "Table meets llm: Can large language models understand structured table data? a benchmark and empirical study." Proceedings of the 17th ACM International Conference on Web Search and Data Mining. 2024.
5. Fernandez, Raul Castro, et al. "How large language models will disrupt data management." Proceedings of the VLDB Endowment 16.11 (2023): 3302-3309.
6. Sharma, Richa, Pooja Agarwal, and Arti Arya. "Natural language processing and big data: a strapping combination." New Trends and Applications in Internet of Things (IoT) and Big Data Analytics. Cham: Springer International Publishing, 2022. 255-271.
7. LeCun, Yann, Koray Kavukcuoglu, and Clément Farabet. "Convolutional networks and applications in vision." Proceedings of 2010 IEEE international symposium on circuits and systems. IEEE, 2010.
8. Hadji, Isma, and Richard P. Wildes. "What do we understand about convolutional networks?." arXiv preprint arXiv:1803.08834 (2018).
9. Ogoke, Francis, et al. "Graph convolutional networks applied to unstructured flow field data." Machine Learning: Science and Technology 2.4 (2021): 045020.
10. Coscia, Dario, et al. "A continuous convolutional trainable filter for modelling unstructured data." Computational Mechanics 72.2 (2023): 253-265.
11. Tang, Ning, et al. "Improving the performance of lung nodule classification by fusing structured and unstructured data." Information Fusion 88 (2022): 161-174.

 doi:10.20944/preprints202407.1364.v1



12. Yu, Fisher, and Vladlen Koltun. "Multi-scale context aggregation by dilated convolutions." arXiv preprint arXiv:1511.07122 (2015).
13. Strubell, Emma, et al. "Fast and accurate entity recognition with iterated dilated convolutions." arXiv preprint arXiv:1702.02098 (2017).