Research on Interpretable Text Sentiment Analysis

Ding Fei

A Thesis submitted to Tokushima University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

March, 2024

Department of Information Science and Intelligent Systems
Graduate School of Advanced Technology and Science
Tokushima University, Japan

Contents

List of Figures

List of Tables

Abstract

With the proliferation of Deep Learning (DL) methodologies and Large Language Models (LLMs) in sentiment analysis, significant advancements have been achieved in recent years. However, these models introduce a set of challenges that have not been adequately addressed. Foremost among them is the lack of model interpretability, which hinders the understanding of the mechanisms through which these models make decisions. Furthermore, the current state-of-the-art models necessitate expensive computational resources for training and require vast datasets with manual annotations, posing both financial and time constraints for researchers and practitioners.

Addressing these challenges, this research presents a unique, interpretable framework for text sentiment analysis that is not only cost-efficient but also high in precision. Our proposed framework synthesizes three principal methods:

1. Enriched Semantic Layer: By merging the unsupervised topic clustering Latent Dirichlet allocation (LDA) matrix with the hidden expression matrix generated by Transformer-like models, we bolster the semantics of the hidden layer. This synthesis allows the model to capture deeper and more nuanced sentiments from the text, bridging the gap between raw data and interpretability.

2. Symbolic Logic Integration: We incorporate symbolic logic systems, such as Real Logic and Logic Tensor Network (LTN), into our framework. By doing so, we translate the traditionally obscure operations of deep learning models into a more understandable and logical format. This layer of logic helps decode the complex operations, rendering the model interpretable to a significant extent.

3. Large Language Model Supervision: Using a sophisticated language model, such as ChatGPT, as the teacher model, we generate target text. This text acts as a benchmark, evaluating the quality of the text produced by the student model. Through this teacher-student Causal Inference dynamic, our framework gains insights from

state-of-the-art models without inheriting their inherent opacity. In this process, we also utilies technologies such as Prefix Instruction Fine-tuning, within-Domain Further Fine-tuning, and Task-specific Decoding to further improve the efficiency and accuracy of the model.

Incorporating these strategies, our methodology prioritizes both model simplicity and transparency, while also leveraging domain knowledge. Initial results indicate that this hybrid approach melds interpretability with high performance, suggesting a compelling alternative to the prevailing deep learning-centric models. This research aspires to spearhead the development of more transparent, efficient, and accessible sentiment analysis tools in the future.

In order to further verify the effectiveness of the framework proposed in this research, we conducted extensive experiments on multiple sentiment analysis subtasks. These include: Weibo emotion detection, emotion intensity analysis, financial argument analysis, human-machine customer service dialogue satisfaction evaluation, etc. The model outperforms state-of-the-art baselines on various subtasks and achieves first place in both NTCIR-16 DialEval-2 and NTCIR-17 FinArg-1 tasks.

Keywords: Affective computing; Text sentiment analysis; Model fine-tuning; Prompt engineering; Feature fusion

1 Introduction

1.1 Motivation

As we dive deeper into the digital age, vast amounts of textual data are generated every second across online platforms, ranging from social media posts and news articles to customer reviews and patient feedback. Understanding the sentiment embedded within this data can provide invaluable insights into consumer preferences, political landscapes, and even predictive healthcare outcomes. The field of sentiment analysis with Artificial Intelligence (AI) has thus rapidly evolved to decode these nuanced emotions and opinions hidden in textual form [1].

However, with the advent and dominance of deep learning models in sentiment analysis, a paradoxical situation has emerged. While these models can achieve impressive accuracy, they often function as "black box" [2], with their intricate architectures and millions of parameters making it challenging to discern how they arrive at particular decisions. This opacity is problematic for several reasons:

1. Accountability & Trust: In sectors where decisions have profound implications, like finance, healthcare, or public policy, understanding the rationale behind a sentiment prediction is crucial. It instills confidence in stakeholders and ensures accountability in AI-driven decisions.

2. Model Improvement: A clear understanding of how a model processes information can highlight areas of improvement. For instance, if a model consistently misinterprets certain phrases or contexts, pinpointing the cause becomes feasible with interpretability.

3. Regulatory Compliance: In many industries, regulations mandate that algorithmic decisions be explainable to those affected by them. An interpretable sentiment analysis model would thus align better with such regulatory landscapes.

4. Generalization & Bias Detection: Interpretable models can help in identifying biases or over-generalizations the model might have inherited from its training data, ensuring fairness and broader applicability.

Moreover, there's an economic dimension to consider. Advanced deep learning models come with heavy computational demands, translating to high training costs [3]. These costs are not just financial—labor cost concerns are emerging regarding the extensive AI training sessions. By focusing on interpretable models, there's potential to develop more efficient algorithms that require less data and computational power. The promise is twofold: reduced costs for businesses and researchers, and more sustainable approach to AI training.

Given the immense potential applications of sentiment analysis in shaping business strategies, informing public policies, enhancing patient care, and more, there's a pressing need for models that are not only accurate but also transparent in their operations. Interpretable text sentiment analysis stands at this crucial intersection, aiming to demystify the complexities of sentiment predictions and make AI more accessible, trustworthy, and effective.

1.2 Significance of Research

The importance of understanding human emotions and sentiments in textual data has always been clear, but with the surging reliance on machine learning and artificial intelligence, the need for interpretable text sentiment analysis has gained newfound urgency. The significance of this research extends across multiple dimensions.

One of the most pressing concerns in AI today is the opacity of decision-making processes, particularly in deep learning models. This research offers a tangible solution, seeking to make intricate AI decisions understandable and relatable to humans, thus bridging this transparency gap. When stakeholders in any domain, be it finance, healthcare, or public policy, can understand the 'why' behind an AI's sentiment prediction, it

augments their decision-making process. With clarity on how conclusions are drawn, decisions can be made confidently and with increased precision. As societies grapple with the ethical implications of AI, regulations are emerging that demand transparency in AI-driven decisions. Research in interpretable sentiment analysis is thus pivotal in ensuring that AI tools remain compliant with evolving legal landscapes, avoiding potential legal ramifications. Simultaneously, with an emphasis on interpretability often comes model simplicity and efficiency. By reducing the complexity and data demands, businesses and researchers can deploy sentiment analysis tools at a fraction of the traditional cost. This democratizes access, allowing even small-scale entities to harness the power of sentiment analysis without exorbitant investments. Last but not least, transparent models can shed light on inherent biases in AI algorithms, which can originate from skewed training data or model architectures. By making these biases visible, this research plays a pivotal role in promoting fairness and inclusivity in AI tools, ensuring that they serve diverse populations equitably.

In summary, the significance of research on interpretable text sentiment analysis cannot be overstated. It promises a future where AI not only understands human sentiments with high accuracy but also communicates its conclusions transparently, operates ethically, and does so in an economically and environmentally efficient manner. This research thus stands at the intersection of technological advancement, ethical considerations, and practical applications, holding the potential to redefine the landscape of sentiment analysis.

1.3 Research Contents and Contributions

Affective computing, first provided by Picard in 1977 [4], is a key technology for sentiment recognition and opinion mining in the field of natural language processing (NLP). It has a wide range of applications in public opinion monitoring [5], business prediction [6], consumer preference analysis [7], patient situation analysis [8], and other fields. As a important branch of affective computing, text sentiment analysis is also an

upstream task of many other NLP tasks, such as style transfer [9, 10], emotional text generation [11], and chatbot [12].

However, text sentiment analysis is a difficult task, as shown in Figure 1.1. Initially, the emotions expressed by words or sentences are often ambiguous or have serious semantic diversity. The phrase "I got you!" may have complex and diverse semantics. In addition, the emotions expressed by the text are usually complex, not single. If we divide emotions into positive and negative, and in some tasks, the positive emotions can be further divided into joy, love, surprise, expect, and the like. The problem is that we have many other complex emotions. This kind of emotion usually contains other emotions, for example, when you miss someone, you always feel expectation and anxiety at the same time. Last but not least, emotions can also be affected by subjective personality. For example, the smiling emoticon is just a smile for the elders. But for young people, it means a lot.

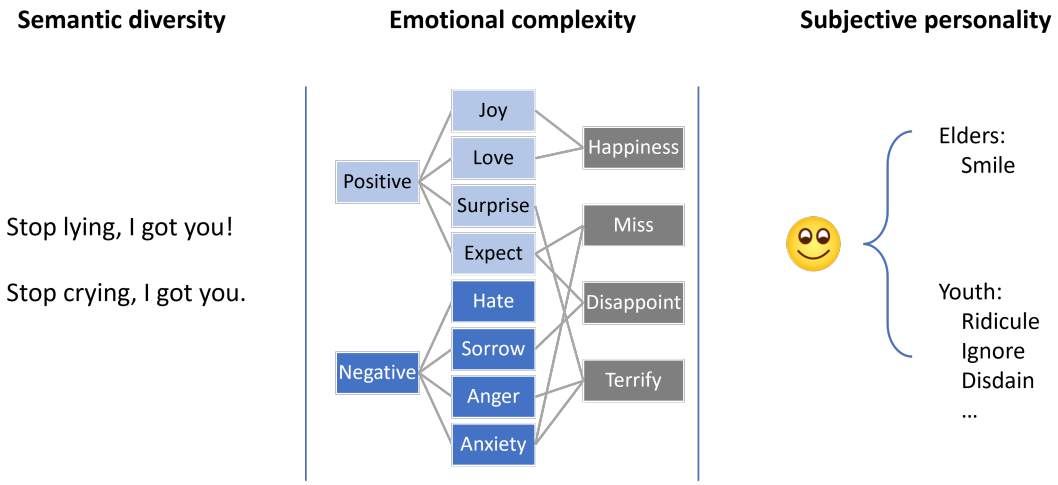

Fig. 1.1. Difficulties of sentiment analysis.

Furthermore, as explained in Sections 1.1 and 1.2, the prevailing sentiment analysis paradigm within the context of large language models is uninterpretable, expensive, and unreliable. To address these challenges, this research introduces a unique and interpretable framework for text sentiment analysis that is both cost-efficient and precise. Our

proposed framework incorporates three primary methods: the Enriched Semantic Layer, Symbolic Logic Integration, and Large Language Model Supervision. Technical details will be elaborated upon in Section 3. When compared to related works, our contributions primarily focus on the following:

1. We introduce a novel architecture for text sentiment analysis that offers reduced training costs, enhanced classification and detection outcomes, and superior interpretability.

2. We investigate the integration of feature vectors from various modalities. This includes feature vectors produced by models with differing supervision types, feature vectors for different levels of text granularity, and feature vectors for multiple languages. The advantages and disadvantages of these combinations are discussed in depth.

3. We examine the potential of merging traditional neural networks with symbolic logic methods in text sentiment analysis tasks. This not only enhances the framework's interpretability but also bolsters its accuracy.

4. We devise a comprehensive set of prompts effective for both LLMs and text generation models. By employing a range of prompt engineering strategies, the text-to-text model becomes adept at text classification and relational reasoning. Furthermore, it can tackle a variety of downstream tasks using simple decoders.

5. We apply the proposed framework to multiple practical sentiment analysis tasks. These include Weibo emotion detection, emotion intensity analysis, financial argument analysis, human-machine customer service dialogue satisfaction evaluation, etc. The model outperforms state-of-the-art baselines on various subtasks and achieves first place in both NTCIR-16 DialEval-2 and NTCIR-17 FinArg-1 tasks.

In summery, this research introduces a unique sentiment analysis framework that addresses challenges in large language models, emphasizing cost-efficiency, precision,

and interpretability. Key contributions encompass the development of a novel sentiment analysis architecture with heightened efficiency and clarity, the innovative integration of feature vectors from various modalities, the pioneering combination of traditional neural networks with symbolic logic methods, and the crafting of versatile prompts optimized for both LLMs and text generation, ensuring superior classification, reasoning, and adaptability to diverse tasks.

1.4 Thesis Organizations

The research on Interpretable Text Sentiment Analysis is divided into five sections in this paper. Each section of this paper is organized as follows:

1. Introduction.

   This section introduces the research background of the affective computing and the significance difficulties of sentiment analysis. Motivation and innovation of our research are also introduced.

2. Background and preliminaries.

   Necessary preliminary knowledge is introduced briefly in this section. Some related or similar works are also listed.

3. Interpretable text sentiment analysis framework.

   In this section, we introduce the proposed model and method of our research, including enriched semantic feature fusion, symbolic logic integration and large language model supervision.

4. Experiments and analysis.

   This section includes detail experiments and discussions. Extensive experimental results demonstrate that our model outperforms state-of-the-art baselines on various subtasks. In addition, ablation experiments and case studies are proposed.

5. Conclusions and future work.

   We conclude our work and outline the direction of future work.

2 Background and Preliminaries

2.1 Text Sentiment Analysis

The goal of sentiment analysis is to extract and analyze information from subjective material, such as blogs and tweets published on the Internet, or conversations in an online customer service system. Sentiment analysis has lately been a popular study subject in data mining and Natural Language Processing (NLP) due to its wide range of academic and corporate applications, as well as the exponential growth of social media data [1]. Sentiment analysis is a challenging process. First, the emotions communicated by words or phrases are frequently imprecise or with a significant semantic variation. Second, the feelings portrayed by the text are frequently complicated, rather than simple. Lastly, subjective personality can have a significant influence on emotions.

Generally speaking, the process of text sentiment analysis can be divided into three parts, as shown in Figure 2.1: text information collection, sentiment feature extraction, and information analysis. The text information collection module obtains the sentiment comment text through text grabbing tools (such as web crawler tools) and transmits it to the sentiment feature extraction module. The sentiment feature extraction module transforms the natural language text into a form that can be recognized and processed by the computer and gives it to the information analysis module to get different results.

1. Text information collection.

   Emotional feature tagging is to annotate emotional semantic features, usually taking words or semantic blocks as feature items. Emotional feature tagging firstly designs the attributes of emotional semantic features, such as commendatory words, derogatory words, enhanced mood, general mood, sadness, happiness and so on; then it labels the emotional semantic features by machine automatic tagging or manual tagging to form a set of emotional features. Emotion dictionary is a typical set of emotion features, and it is also the basis of emotion computing. In most researches, the research on affective computing usually introduces the affective dictionary di-

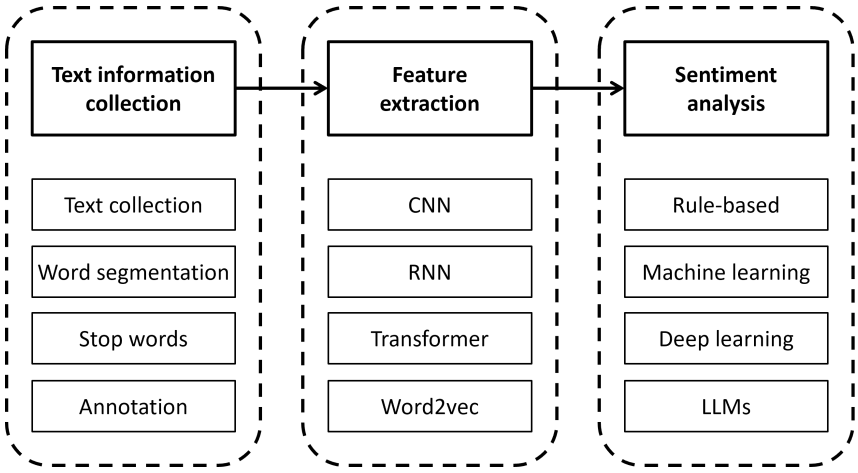

Fig. 2.1. General process of sentiment analysis.

rectly into the custom dictionary.

It is a simple and rapid method to calculate the text emotion value with an emotion dictionary, but the accuracy needs to be improved. In the actual emotional computing, it will be different because of the specific language application environment. For example, the word "sick" is usually regarded as a negative word, but it is regarded as a positive word in spoken English when used with “pretty”. At the same time, there are often negative preposition, double negation, colloquialism, and expression used in text, which will have a great impact on the extraction and judgment of text emotional features. Therefore, when extracting text emotion, we need to analyze the text and its corresponding context and environment.

2. Sentiment feature extraction.

   The text usually contains complex emotional information. In the research of giving computers the ability to recognize text emotion, it is very important to extract feature patterns from text signals. After preprocessing the text, we need to extract emotional semantic features. The basic idea of feature extraction is to determine which features can give the best emotion recognition according to the text data. The usual algorithm is to score the existing emotional feature words, and then the fea-

ture subset is composed of the features that exceed a certain threshold in the order of score. The quality of the feature word set directly affects the final result, so in order to improve the accuracy of calculation, the research of text feature extraction algorithm will continue to be concerned. In the long run, the technology of automatic text feature generation will be further improved, and the research focus of feature extraction will shift from the analysis of word frequency to text structure and emotional words.

3. Emotion information analysis.

   In the early text sentiment analysis technology, there are mainly two technical routes: rule-based method and statistics-based method. In the 1980s, the rule-based approach dominated the mainstream position. It acquired syntactic rules through linguists' language experience and knowledge and used them as the basis of text analysis. However, the process of acquiring rules is complex and costly, which also has a negative impact on the performance of the system, and it is difficult to find an effective way to improve the efficiency of developing rules. After the 1990s, people tend to use statistical methods to select features and train parameters through training samples. According to the selected features, the input samples to be classified are formalized, and then input to the classifier to determine the category, and finally get the category of the input sample. Nowadays, with the rise of LLMs, more and more companies and researchers prefer to use LLMs for sentiment analysis.

Recently, many sentiment analysis problems, such as polarity detection [13] and sentiment intensity prediction [14], have been effectively handled using traditional machine learning approaches including Support Vector Machines (SVM) [15], Latent Dirichlet allocation (LDA) [16], and graph mining. More recently, Deep Neural Networks (DNN) have demonstrate their impressive ability to imitate the hierarchical organizational structure of the human brain, enhancing their capacity for deeper emotional semantic expression capabilities, and have been widely employed in the field of sentiment analysis. In

particular, Transformer-based language models [17, 18], i.e., deep neural networks that are first pre-trained on massive corpora and then fine-tuned for a specific domain, are the emerging paradigm that successfully tackles the sentiment analysis tasks.

Although DNN networks achieve human-like accuracy on many tasks, they are not without flaws. Most DNN models require a substantial amount of annotated data and significant training costs, and lack interpretability due to their "black box" nature. Over the past few decades, many researchers have attempted to solve the aforementioned problems by combining traditional computing approaches and DNN models. On the traditional computing side, it is desirable to retain the high interpretability and provability inherent in these systems, as well as the simplicity of leveraging expert human background knowledge. On the DNN side, desirable advantages include trainability on raw data and robustness to faults in the underlying data. As a result, methods like Symbolic or Subsymbolic AI for sentiment analysis have emerged [19, 20, 21, 22]. These methods are committed to integrating logical reasoning within deep learning architectures to build a commonsense knowledge base for sentiment analysis.

2.2 LDA for Topic Clustering

LDA (Latent Dirichlet Allocation) [16] was proposed by Blei, Ng and Jordan in 2003 to infer the topic distribution of documents. The LDA model is a traditional three-layer Bayesian model. The core concept is to describe both the document-topic and the topic-word as a Multinomial distribution with a Dirichlet prior probability. LDA employs hyperparameters in the parameter settings to solve the overfitting problem in other approaches such as PLSA [23]. The LDA model utilized in this paper is a three-level generative model, as illustrated in Fig. 2.2. The document is viewed as a sequence of N words, and the corpus is made up of D documents. There are K subjects in the corpus, thus K signifies that there are K topics in the corpus. The boxes in the figure represent repeated sampling.

The idea of topic word is not taken into account in the standard unigram model.

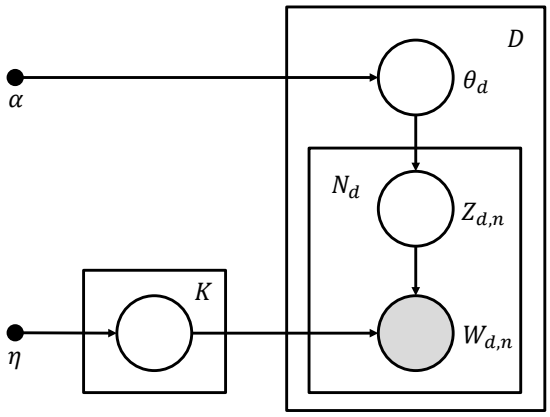

Fig. 2.2. The basic LDA architecture.

When we produce articles, we make absolutely sure that each content is about a certain topic. As a result, in the LDA structure depicted in Fig. 2.2, each circle represents a random variable. The grey circles are the observable variables, while the white circles are the latent variables. Words in the corpus are the only information we can observe in actual activities. The goal of LDA is to use observable words to infer the underlying topic information. When constructing a text corpora model, the generative procedure is as follows (assuming the corpus comprises D documents and K topics):

1. For each topic $k \in K$, calculate $\beta_k \sim \text{Dirichlet}(\eta)$. This draws a distribution of the words, which can be treated as the probability of a word appearing in topic k.
2. For each document $d \in D$, calculate the topic proportions $\theta_d \sim \text{Dirichlet}(\alpha)$.
3. For each word i in document d:
   (a) Calculate the topic assignment $z_{di} \sim \text{Multinomial}(\theta_d)$.
   (b) Calculate the observed word $w_{ij} \sim \text{Multinomial}(\beta_{z_{di}})$.

The bag of words model is also utilised in LDA. D documents correspond to d independent Dirichlet-Multinomial conjugate structures, whereas K subjects are k independent Dirichlet-Multinomial conjugate structures. Each word in a document must be generated by rolling the dice twice. The first time we roll a doc-subject dice and receive a topic; the

second time we roll a topic-word dice to get a word. Each time we generate a word in each document, the two actions of rolling the dice are carried out close rotational. If the corpus contains N words, we will roll the dice $2N$ times. In this scenario, we can obtain the following outcomes:

$$\begin{aligned} p(w,z \mid \alpha,\eta) &= p(w \mid z,\eta)p(z \mid \alpha) \\ &= \prod_{k=1}^{K} \frac{\Delta(\beta_k+\eta)}{\Delta(\eta)} \prod_{d=1}^{D} \frac{\Delta(\theta_d+\alpha)}{\alpha}. \end{aligned} \tag{2.1}$$

The classic topic generating model LDA has a strong mathematical foundation and flexible scalability, and it has produced some great topic extraction performances. However, because LDA ignores the local subject words, it cannot directly examine the subject's emotional polarity. The LDA approach of text topic extraction is an unsupervised method, and the number of topics K is a hyperparameter. The value of K has a significant impact on the performance of sentiment classification with diverse text lengths. If K is too small, the extracted topic expression will have inadequate weight in comparison to the Transformer hidden representations, resulting in a diluted extracted probability. On the other hand, if K is too big, the extracted topic expression will be excessively intricate, resulting in overfitting and loss of topic information. In this research, we experimented with K values ranging from 5 to 50, and we treated text as a "text sequence" at different levels: sentence, paragraph, or document. Section 4 contains more information on the parameters and settings.

2.3 Logic Tensor Network

Logic Tensor Network (LTN) [24, 25] is a Neural-Symbolic (NeSy) framework that supports the learning of neural networks by using the satisfaction of a first-order logic knowledge base as an objective. In other words, LTN employs logical reasoning on the knowledge base to guide the learning of a potentially deep neural network. The concept behind LTN is straightforward: it consists of a first-order logic knowledge base

containing a set of axioms; some predicates, functions, or logical constants appearing in these axioms are targeted for learning; and there is data available that can be utilized to learn the parameters of these symbols. The approach involves using the logical axioms as a loss function for the Logic Tensor Network. The objective is to find solutions in the hypothesis space that maximally satisfy all the axioms contained in the knowledge base.

In LTN, the learnable parameters are contained in the predicates, functions, and possibly learnable logical constants that appear in the logical axioms of the knowledge base. The LTN framework can be easily implemented through the Pytorch tool using LTNtorch [26]. The background knowledge that LTN can express can be roughly divided into three categories:

1. **Symbol embedding**: including domain boundaries, explicit expressions of symbols, and parameterized definitions of symbols;
2. **Formulas**: including factual propositions and generalized propositions;
3. **Fuzzy syntax**: implemented by defining operators.

Through the above methods, the LTN framework can complete a variety of tasks, such as: Binary Classification, Multi-class Single-label Classification, Multi-class Multi-label Classification, Semi-supervised Pattern Recognition, Regression, Clustering, or Learning Embeddings.

Here gives a workflow that uses the LTN framework to complete Binary Classification tasks, as shown in Fig2.3. In the figure, $G(x_+)$ and $G(x_-)$ are inputs to the network $G_\theta(A)$ and the dotted lines indicate the propagation of activation from each input through the network, which produces two outputs. Suppose that one wants to learn a binary classifier A for a set of points in $[0,1]^2$. Suppose that a set of positive and negative training examples is given. LTN uses the following language and grounding:

Domains:

points (denoting the examples).

Variables:

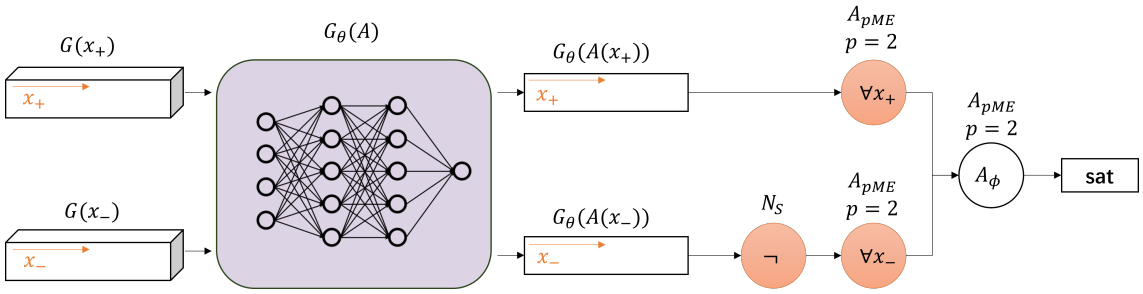

Fig. 2.3. Symbolic tensor computational graph for the binary classification example.

x_+ for the positive examples;

x_- for the negative examples;

x for all examples;

$D(x) = D(x_+) = D(x_-) = points$, where $D(.)$ is a function which returns the domain of a non-logical symbol given in input.

Predicates:

$A(x)$ for the trainable classifier;

$D_{in}(A) = points$, where $D_{in}(.)$ is a function which returns the domain of the input of a function or predicate given in input.

Axioms:

$\forall x_+ A(x_+)$: the predicate must be true for positive examples;

$\forall x_- \neg A(x_-)$: the predicate must be false for negative examples.

Let us define E the data set of all examples. The objective function with $K = \{\forall x_+ A(x_+), \forall x_- \neg A(x_-)\}$ is given by $SatAgg_{\phi \in K} G_{\theta, x \leftarrow E}(\phi)$. In practice, the optimizer uses the following loss function:

$$L = 1 - \underset{\phi \in K}{\text{SatAgg}}\, G_{\theta, x \leftarrow B}(\phi), \tag{2.2}$$

where B is a mini batch sampled from E. By minimizing the loss, we will try to maximize the SatAgg operator applied to the knowledge base. Maximizing the SatAgg operator

means maximizing the satisfaction level of each formula included in the knowledge base.

In general, LTN achieves an interpretable training process by replacing the loss function in the ordinary deep learning gradient descent process with a knowledge base composed of first-order logic and real logic. Theoretically speaking, model $G_\theta(A)$ in Fig. 2.3 can be any neural network model. In this study, we mainly tested on the BERT base model and Multi-label Multi-class Emotion Classification tasks. Detailed grouding process can be found in Section 3.

2.4 Languege Models

2.4.1 Transformer based Models

Transformer architecture, as shown in Fig. 2.4, was first proposed by Google in "Attention is all you need" [27] on machine translation task in 2017, which caused considerable repercussions.[1] Transformer's main mechanism is known as Self-attention, which converts the distance between two words in any location to one using three new vectors. These three vectors are referred to as Query, Key, and Value, respectively. The entire procedure is as follows:

1. Firstly, Self-attention generates three additional vectors, Query, Key, and Value, with a dimension of 512. These three vectors are calculated by multiplying an embedding vector by a randomly initialized matrix, the dimension of which is (64, 512).

2. Then we compute the Self-attention score, which indicates how many attention we pay to the rest of the input sequences when encoding a word in a certain place. The dot product of Query and Key yields this fractional value.

3. The result of the dot product is divided by a constant, which is typically the root of the first dimension of the aforementioned matrix, i.e. 8. Then we do a softmax

[1] Note that the Transformer architecture mentioned in this paper is not what the original paper refers to. The Transformer mentioned in the original paper refers to the complete Encoder-Decoder framework, and the Transformer architecture mentioned in this paper can be simply understood as the Encoder part in the original Transformer.

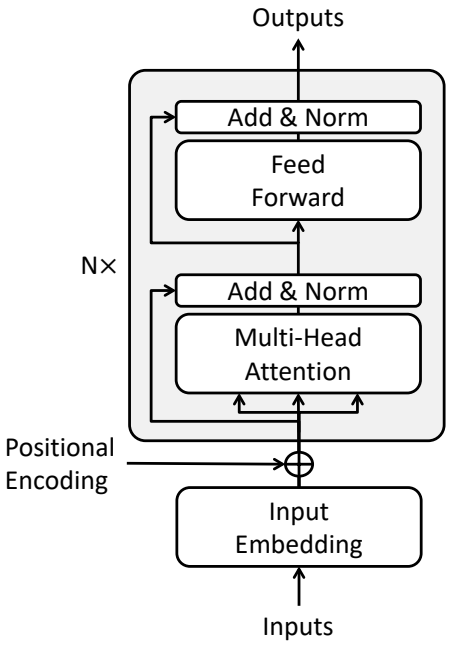

Fig. 2.4. The transformer model architecture. Gray background part represents a transformer block.

calculation of the result. The final result is the correlation of each word in a certain position. Additionally, the word correlation value of the current position will certainly be large.

4. Then the value obtained from the previous step is multiply and add by softmax, and the end result is the score of Self-attention in the current position.

In the actual application case, we utilise a matrix to compute using GPU to boost computation performance. Self-attention can be represented as:

$$\text{Attention}(Q,K,V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{2.3}$$

where Q, K and V present the Query, Key and Value matrix respectively. Self-attention differs from the classic seq2seq paradigm in the following two ways. One advantage is that the encoder sends more data to the decoder. The encoder will represent the hidden state of all tokens regarding the decoder, not simply the hidden state of the encoder's final token. Another difference is that the decoder does not simply accept the hidden

state represented by all encoders as input, but instead uses a selection method to pick the most appropriate hidden state for the current position.

To boost the model's performance, the original Transformer not only initializes one group of Q, K, V matrices, but eight. So the ultimate output is calculated by eight matrices, which is called Multi-head Self-attention. The Multi-head Self-attention can be expressed as:

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\underset{1}{\text{head}},\ldots,\underset{h}{\text{head}})W^O, \tag{2.4}$$

where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$. Each sublayer in the original Transformer model is followed by a feedforward module and a normalisation layer. There are several types of normalisation, but they all serve the same purpose: to turn the input data into output with a mean value of 0 and a variance of 1.

Due to the fantastic results achieved by the Transformer model on multiple NLP tasks in recent years, various Transformer-based methods are constantly being proposed. Representative models are BERT [28], Roberta [29], GPT-2 [30], XLM [31], etc. Transformer models today have a large family and can be easily used via Huggingface [32].

2.4.2 Large Languege Models

Large Language Models (LLMs) [33] represent a significant leap in the field of artificial intelligence, particularly in natural language processing (NLP). These models, based on deep learning techniques, are trained on vast datasets and are capable of understanding, generating, and translating text in a way that closely mimics human language. Their performance has been revolutionary, setting new benchmarks in tasks like text completion, question answering, translation, summarization, and more.

The development of LLMs has been characterized by rapid advancements and a steady increase in the size and complexity of the models. Starting with models like GPT and BERT, the field has seen an exponential growth in the scale of these models. GPT-3 [34], developed by OpenAI, is one of the most well-known examples, featuring 175

billion parameters. Its successor, GPT-4 [35], and others like Google's T5 [36], or Facebook's BART [37], continue this trend, pushing the boundaries of what's possible with AI in understanding and generating human-like text.

Taking the T5, or "Text-To-Text Transfer Transformer" model as an example, it is a notable advancement in the field of natural language processing (NLP) developed by Google. Unlike traditional models that are designed for specific tasks such as translation, summarization, or question answering, T5 adopts a unified framework where every NLP task is converted into a text-to-text format. This means that inputs are always treated as text and the outputs are also generated as text. For instance, a translation task is framed as converting a sentence in one language to another, while a classification task involves transforming a piece of text into a label. The T5 model is pre-trained on a large corpus of text in a self-supervised manner, using tasks like "masked language modeling" where the model predicts missing words in a sentence. This pre-training helps the model understand language context and structure. It is then fine-tuned on specific tasks, allowing for impressive flexibility and performance across a wide range of NLP applications.

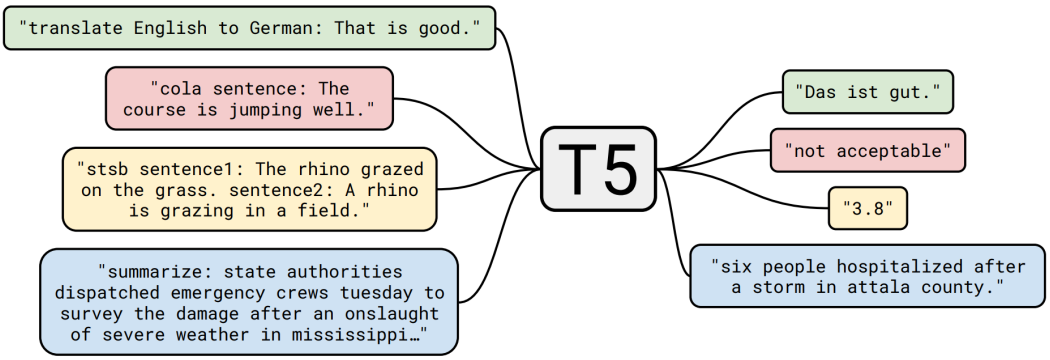

Fig. 2.5. The workflow of T5.

T5 works well on a variety of tasks out-of-the-box by prepending a different prefix to the input corresponding to each task, as shown in Fig. 2.5. For example, consider an English to German task, the original input is:

```
{"en": "That is good.", "de": "Das ist gut."}
```

Now it transform to:

```
{"inputs": "translate English to German: That is good.",
"targets": "Das ist gut.’’}
```

Large language models have substantially enhanced various NLP tasks. Their advanced capabilities in text generation and completion enable them to produce coherent, contextually appropriate, and creative outputs. In question answering, these models excel in providing precise and relevant responses by effectively processing large data sets. LLMs have improved sentiment analysis and text classification, offering deeper insights into emotions and categorizations in texts. Furthermore, they have refined the accuracy in tasks like named entity recognition and part-of-speech tagging, and provides the possibility of automatically generating causal inference for logical reasoning.

2.5 Related Works

As a key branch of affective computing [38], text sentiment analysis offers a wide range of applications in consumer preference analysis [7], public opinion monitoring [5], patient situation analysis [8], business prediction [6], and other fields. It is also an upstream task of many other NLP tasks, such as style transfer [9, 39], emotional text generation [40], and cross-domain learning [41, 42]. Many text sentiment analysis algorithms have been recently developed, based on the “dictionary + rule” [43, 44, 45, 46] method, machine learning [47, 48, 49], or deep learning [50, 21]. Traditional sentiment analysis methods confront several difficulties and challenges. Early methods, reliant on the creation and adaptation of a dictionary or traditional machine learning approaches, often exhibit low domain adaptability. Furthermore, a single sentiment dictionary struggles to manage complex ambiguity situations, leading to these approaches typically being used as supplementary methods in conjunction with other procedures. While deep learning circumvents the limitations of advanced feature engineering compared to machine

learning approaches, supervised deep learning still necessitates a large annotated corpus for training. At the same time, unsupervised deep learning has strict requirements for semantic correlations in data and requires additional development.

Following the Transformer based model's astounding success in numerous NLP tasks, the "pre-training + fine-tuning" framework paradigm has gained popularity among more and more researchers. Recently, many "pre-training + fine-tuning" models are employed in sentiment analysis tasks. Sun et al. [51] propose a general solution to fine-tune the pre-trained BERT model on text classification tasks and target task. We employ pre-traind BERT and topic clutering to solve the multi-label emotion classification problem [52]. On the basis of this research, we conduct additional experiments to verify the correlation between the selection of Transformer model, fine-tuning method, feed-forward network architecture, the setting of the number of topics and the final classification effect in this paper. Huang et al. [53] proposed an ensemble approach composed of two deep learning models (LSTM and BERT) for contextual emotion detection task. Zhang et al. propose an Efficient Adaptive Transfer Network (EATN) for aspect-level sentiment analysis which emphasizes the need to incorporate the correlation among multiple domains [54]. Other methods, such as DATN [55], HATN [41] or IATN [42], also exploit the combination of transfer network and pre-trained model to solve the Cross-Domain Sentiment Classification task. These pre-trained Transformer-based models have shown to be quite powerful in the field of sentiment analysis. However, they have certain drawbacks, such as a lack of reasoning and interpretability, difficulties in training from small datasets, and poor transferability to other domains.

Another prevalent paradigm for enhancing sentiment analysis outcomes is to combine traditional logical methods with deep neural methods. Early neuro-symbolic AI was applied to vision and language understanding task, yielding breakthrough progress. Yi et al. propose NS-VQA and its improved system [56, 57], which combines two strong concepts: deep representation learning for visual recognition and language comprehension, and symbolic program execution for reasoning. This system using symbolic structure

as previous knowledge has distinct advantages. Hamilton et al. [58] conducted a structured review of studies implementing neuro-symbolic for NLP, challenges, and future directions, with the goal of determining whether neuro-symbolic is meeting its promises of reasoning, out-of-distribution generalization, interpretability, training from small data, and transferability to new domains. Cambria et al. [19] integrate top-down and bottom-up learning via an ensemble of symbolic and subsymbolic AI tools to solving sentiment analysis tasks. Huang et al. [59] proposed a combination model of LDA and the Internet short review theory to solve the problem of sentiment analysis on Internet short text, which is called the TSCM model. State-of-the-art research on neuro-symbolic AI is focusing on more fine-grained sentiment analysis tasks. For example, SKIER [60] leverages different symbolic knowledge graph relations to learn knowledge-enhanced features for the Emotion Recognition in Conversation (ERC) task. Zeroc [61] introduces a neuro-symbolic architecture that can recognize and acquire novel concepts in a zero-shot manner. Concurrently, many frameworks and tools for neuro-symbolic method have been proposed, such as SenticNet 7 [62] and PyReason [63]. The emergence of these frameworks has expedited the study of neuro-symbolic AI in sentiment analysis, facilitating faster and easier research while also presenting significant opportunities and challenges.

As NLP models enter the era of large models, more complex domain-specific tasks are constantly breaking records. In terms of applications in specialized fields, AI technology has also garnered significant attention from researchers in areas such as healthcare [64, 65], education [66, 67], smart home [68, 69], and finance [70, 71]. Interestingly, many of these studies highlight the interpretability issues and the high training costs associated with current large-model AI technologies. As one of the pioneering LLMs, T5 [36] is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks converted into a text-to-text format. It offers a unified framework for the realm of NLP pre-trained models by unifying diverse tasks into a single format. Subsequent research such as [72, 73] explored the limits of text-to-text generative models applying to the Aspect-Based Sentiment Analysis (ABSA) tasks [74]. Jordan et al. [75]

introduced a Chatbot Interaction with Artificial Intelligence (CI-AI) framework which outlines a strategy for training transformer-based, chatbot-esque architectures for task classification. Within this approach, T5 plays a pivotal role in data augmentation. Similarly, a study by [76] explored a framework for fact verification. This framework harnesses pre-trained sequence-to-sequence transformer models and employs T5 in a listwise method, paired with data augmentation techniques.

As one of the representative tasks in a specific field, Financial Argument Analysis tasks have attracted more and more attention from scholars. Financial information is inherently dynamic. BloombergGPT [77] retrains an LLMs using a mixed dataset of finance and general data sources. This endeavor consumed approximately 1.3M GPU hours, translating to a staggering cost of around $5M. Given the prohibitive expenses associated with retraining LLMs on a monthly or even weekly basis, there's a pronounced preference for more lightweight adaptations within the finance sector. In response, [78] unveiled an interpretable neural network framework tailored for financial analysis. This solution adopts a hierarchical strategy, complemented by a query-driven attention mechanism, to discern sentiments in financial news texts. In a similar vein, Dogu Tan Araci [79] introduced FinBERT, a language model rooted in BERT, designed to address nuanced tasks specific to the financial landscape. Remarkably, even with a condensed training dataset and by fine-tuning only segments of the model, FinBERT consistently surpasses the performance benchmarks set by leading machine learning methodologies. Keane Ong et al. proposed FinXABSA [80], a novel approach for enhancing explainability in financial analysis. This technique employs the Pearson correlation coefficient to draw connections between aspect-based sentiment analysis and stock price fluctuations. In a similar vein, Hongyang Yang et al. present an open-source large language model named FinGPT [3], tailored for the finance domain. Setting it apart from proprietary counterparts, FinGPT champions a data-centric ethos, offering both researchers and industry professionals a transparent and readily available resource to evolve their financial LLMs. The contemporary trend in the NLP sector revolves around leveraging robust LLMs as

foundational models. These are further refined through fine-tuning, data augmentation, Parameter-Efficient Fine-Tuning (PEFT) [81], and other techniques to address intricate downstream challenges.

2.6 Conclusion

In conclusion, section 2 has laid a comprehensive foundation for understanding key concepts and methodologies that are pivotal in the field of Interpretable Text Sentiment Analysis. We began by exploring Text Sentiment Analysis, delving into its significance and methodologies for interpreting emotions and opinions in text. This was followed by an examination of LDA for Topic Clustering, highlighting its utility in uncovering latent thematic structures within large text corpora.

The section further introduced the Logic Tensor Network, an innovative approach combining logical symbolic reasoning with sub-symbolic machine learning, offering a novel perspective on handling complex data. In the Language Models subsection, we delved into Transformer-based Models, underscoring their revolutionary impact on NLP tasks through advanced architectures that emphasize parallel processing and attention mechanisms. Lastly, the discussion on Large Language Models provided insights into the latest developments in this domain, showcasing how these sophisticated models, with their immense scale and advanced training techniques, have set new standards in text generation, understanding, and language translation.

3 Interpretable Text Sentiment Analysis Framework

3.1 Introduction and Overview

This work introduces a unique sentiment analysis framework that addresses challenges in large language models, emphasizing cost-efficiency, precision, and interpretability. The overview of the proposed framework is shown in Fig. 3.1.

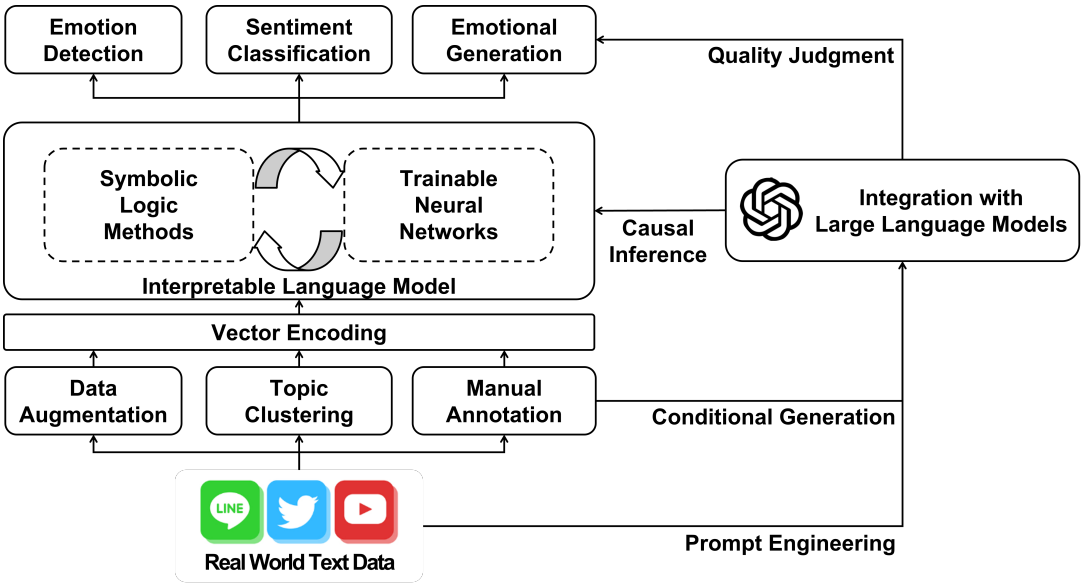

Fig. 3.1. The overview of the proposed interpretable text sentiment analysis framework.

In the era of the information age, we are witnessing an unprecedented explosion of text data, generated across various digital platforms like Twitter, YouTube, and numerous others. This vast expanse of data holds a wealth of information, particularly in terms of emotional content. This is where efficient text sentiment analysis methods become crucial.

Data from various domains and platforms require normalized data cleaning. The semantics embedded in this data can be enriched using methods such as topic clustering and data augmentation. For more details, refer to Section 3.2. In the backbone of the framework, we integrate state-of-the-art transformer-based language models with neural-symbolic networks. This combination allows the framework to maintain the high inter-

pretability and provability of symbolic logic inherent in these systems, as well as the ease of incorporating expert human knowledge. Additionally, it preserves the trainability of neural networks on raw data and their robustness against faults in the underlying data. Further information can be found in Section 3.3. Various pre-training and fine-tuning techniques are applied to the learning process within the framework. Concurrently, a large language model is utilized as a 'teacher model' to iteratively optimize the framework, enhancing the model's capabilities in explanation, reasoning, and causal inference. Detailed insights are provided in Section 3.4. Finally, the framework completes different sentiment analysis tasks through different Task-specific Decoding.

3.2 Enriched Semantic Feature Fusion

3.2.1 LDA Topic Clustering Matrix and Text Feature Extraction

In this section, we will take Weibo Multi-label Emotion Prediction (MEP) task as an example to explain how to integrate topic clustering and feature extraction. General architecture of the proposed method is shown in Fig. 3.2.

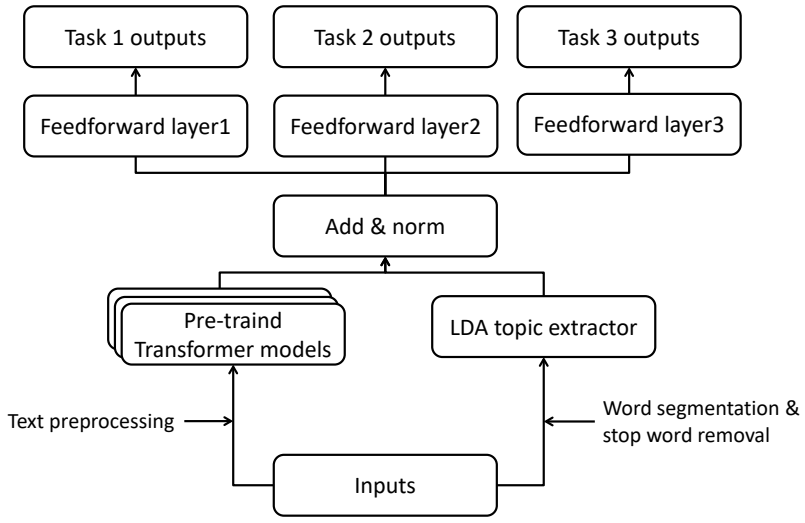

Fig. 3.2. General integrate architecture of LDA topic clustering and text feature extraction method.

The proposed method is applied to Ren_CECps[82] corpus, a well-known Chinese blog sentiment dataset. The corpus was annotated at three different levels: sentence, paragraph, and document. Every level is not only labeled with eight emotional categories (joy, hate, love, sorrow, anxiety, surprise, anger, expect), but also with an emotional intensity rate ranging from 0.0 to 1.0. The entire corpus consists of 1486 Weibo articles with 36525 sentences. In this Multi-label Emotion Prediction (MEP) task, we employ a similar structure like Fig. 3.2, as shown in Fig. 3.3. For better comparative experiments, we use bert-base-chinese as the Transformer model for feature extractor.

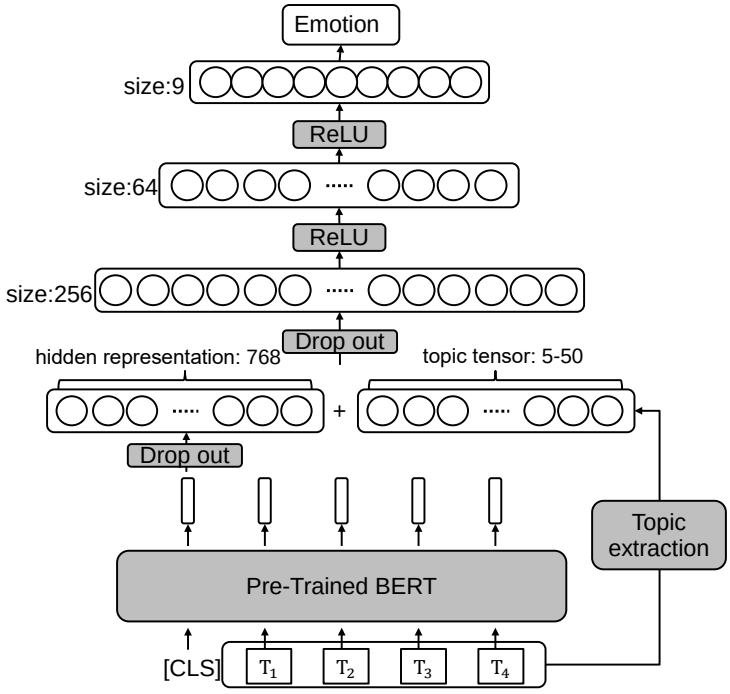

Fig. 3.3. Model detail architecture for multi-label emotion prediction task.

Here gives an example of the topic matrix clustered by the LDA method, as shown in the Fig. 3.4. We set the number of topics to 20 and the clustering level to paragraphs. The table lists the corresponding English translations of the five most frequent words in each topic.

The LDA approach to text topic extraction is an unsupervised method, and the number of topics K is a hyperparameter. The value of K significantly impacts the performance

Topic	Topic Words				
1	Teacher	Friend	Know	Like	Time
2	Know	Now	Like	Beautiful	Always
3	Party	Together	China	Feel	Know
4	Life	Really	Like	Resist	Once
5	Love	Happiness	Life	Joy	Life
6	Know	Teacher	Wish	Child	Japan
7	Man	Woman	Like	Always	Life
8	Know	Now	Feel	Really	Time
9	Thing	Maybe	Like	Sad	Know
10	See	Japan	Daughter	Friend	Then
11	Woman	Like	Some	Man	Look
12	Life	Work	Now	Face	Unable
13	Mom	Know	Phone	Today	Husband
14	Woman	Child	Daughter	Mom	Problem
15	China	Olympic	Game	World	Beijing
16	Now	Feel	Cherish	People	Every
17	China	Country	Great	People	Russia
18	China	Thing	Today	Someone	School
19	Lonely	Know	Already	Miss	Life
20	Some	Like	Friend	Once	Time

Fig. 3.4. Topic clustering example with the total topic number of 20 and clustering level of the paragraph on Ren_CECps.

of sentiment classification across different text lengths. If K is too small, the weight of the extracted topic expression compared to the Transformer's hidden representations may be insufficient, resulting in a diluted probability extraction. Conversely, if K is too large, the extracted topic expression will be overly complex, potentially leading to overfitting and loss of topic information. In this study, we experimented with K values ranging from 5 to 50, and we treated text at different granularity levels: sentence, paragraph, and document. Additional information about the parameters and settings can be found in Section 4.

The maximum input length for bert-base-chinese is 512 tokens, hence the model is incapable of directly reading longer inputs. LDA, on the other hand, does not have this issue. As a result, we tested with six different combinations of BERT levels in sentences and paragraphs, as well as LDA levels in sentences, paragraphs, and documents, respectively. The hidden representations retrieved by BERT and the topic representations extracted by LDA must be fused. There are two issues to consider: how to combine the

hidden representations and how to cope with texts of varying levels.

The LDA model receives a sequence of words as input, from which stop words are deleted and word segmentation is performed. The output of LDA is the topic word probability distribution of K topics and the probability tensor of each input sequence belonging to each topic. We concatenate each topic tensor with the hidden representations extracted by BERT and extend the K-dimensional tensor to one dimension. After concatenation, the size of each tensor ranges from 773 to 818.

Since the maximum input of BERT is 512 tokens, when we input text at the paragraph and document levels, the majority of the sequences surpassed the limit permitted by the BERT model. To cope with the long text problem, we propose the head-tail technique and the hierarchical method, as seen in Fig. 3.5.

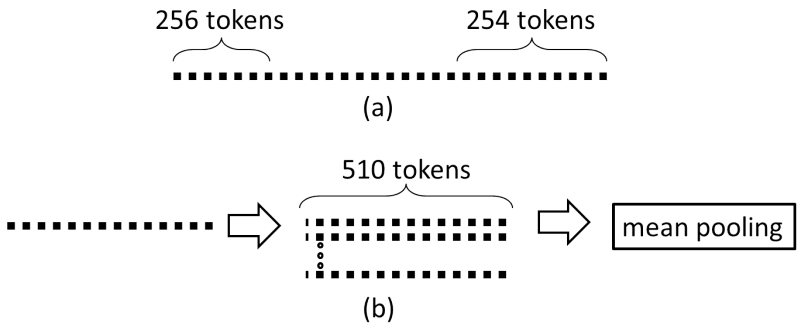

Fig. 3.5. Two methods to deal with the long text problem: (a) Head-tail, (b) Hierarchical.

1. Head-tail: keep the first 256 and the last 254 tokens of the input sequence (the 512 tokens include [CLS] and [SEP], so there are 500 tokens remaining).

2. Hierarchical: the input text is firstly divided into $n = L/510$ fractions, where L represents the length of the input sequence. Then we fed them into BERT to obtain the representation of the n text fractions. Then we use mean pooling to combine the representations of all the fractions.

Upon experimental comparison, the two methods for processing long texts had minimal impact on the results. This is partly because long texts, those exceeding 512 tokens,

make up a small portion (about 1.2%) of the training set. Additionally, the beginning and end of a paragraph often encapsulate sufficient information. Given the constraints on the length of this paper, we won't delve into detailed experiments of these two methods. By default, all subsequent experiments employ the hierarchical long text method.

3.2.2 Parallel Text Translation and Sentence Rearrangement

In this section, we will take human-machine customer service dialogue satisfaction evaluation task as an example to explain how to deal with the Parallel Text Translation and Sentence Rearrangement process. It is worth noting that this process can also be combined with the topic clustering and feature extraction method proposed in Section 3.2, which detailed model overview is shown in Fig. 3.6.

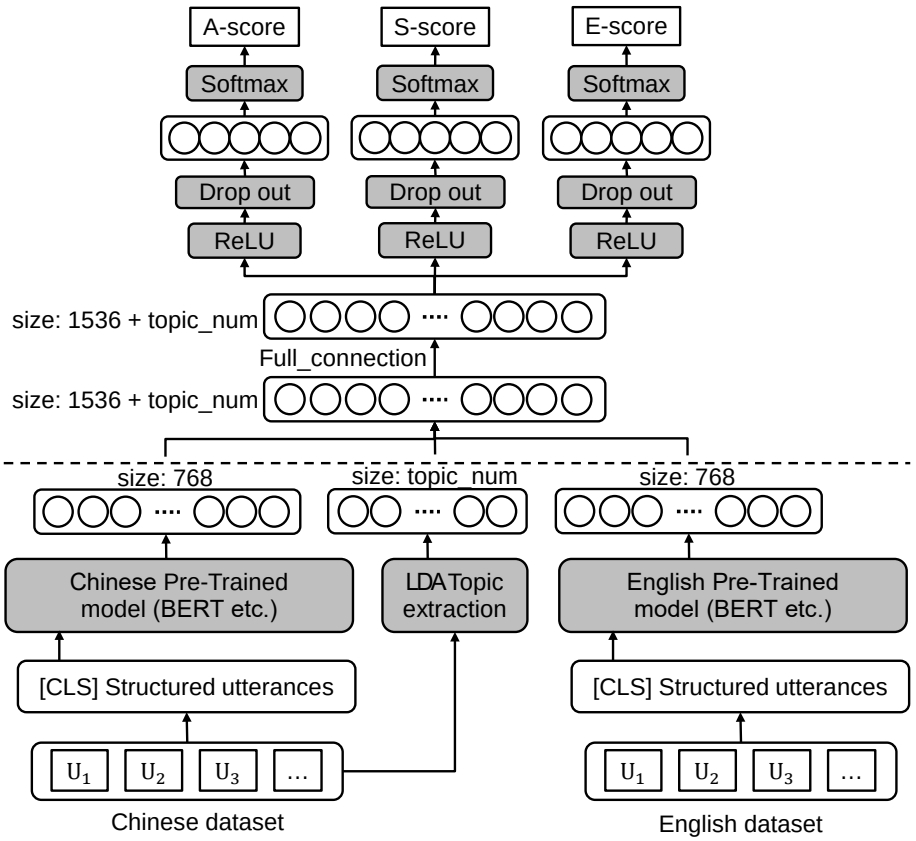

Fig. 3.6. The structure of dialogue quality prediction network. A-score, S-score, E-score present task accomplishment, customer satisfaction, and dialogue effectiveness, respectively.

Human-machine customer service dialogue satisfaction evaluation task has two subtasks: Dialogue Quality prediction (DQ) and Nugget Detection (ND). In the DQ subtask, we examine the proposed method on the DCH-2[83] dialogue corpus for dialogue quality prediction. The DCH-2 corpus contains 4,390 real customer-helpdesk dialogues in Chinese with their English translations. DCH-2 additionally includes dialogue-level and turn-level annotations acquired independently from 19 or 20 annotators for three different sorts of dialogue quality scores.

1. **A-score**: task accomplishment, representing whether a customer's problem has been solved.

2. **S-score**: customer satisfaction, representing whether a customer is satisfied at the end of the dialogue.

3. **E-score**: dialogue effectiveness, representing whether the helpdesk and the customer interact effectively to solve the problem.

The above three scores are annotated from -2 to 2, representing strongly disagree, somewhat disagree, neither agree nor disagree, somewhat agree, strongly agree, respectively. S-score is a standard sentiment polarity annotation, representing the customers' five levels from dissatisfaction to satisfaction. The A-score and E-score have a strong correlation with customers' satisfaction, which plays an auxiliary role in researching users' mental state, and further analyze the sentiment of customers.

We employ multiple pre-trained Transformer models and a LDA topic clustering model, as a feature extractor. The extractor extracts the hidden representations and topical information from the one-to-one translated Chinese and English dialogues respectively. Then we feed the extracted features into a feedforward network to finally predict the quality scores of the input dialogue sequences. Our network simultaneously evaluates three types of dialogue quality, namely the task accomplishment score, the customer satisfaction score, and the dialogue effectiveness score. The two parts are shown in Fig. 3.6, separated by a dotted line in between.

Firstly, in order to better represent the structure of a set of dialogue, we preprocess the input dialogue sequence. For example, consider a tokenized dialogue below:

```
"id": "3830772740080373"
[CLS]
Customer(1/6): "What's going on with ... "
Helpdesk(2/6): "Hi, I'm Little @ of ... "
...
Helpdesk(6/6): "Dear, please choose ... "
[SEP]
```

We add [CLS] and [SEP] special tokens at the beginning and end of each dialogue respectively to fit the training of Transformer models. Moreover, we append meta data at the beginning of every utterance in the dialogues. The meta data consists of both the sender information and the utterance position information in the entire dialogue. Customer/Helpdesk indicates the sender of the utterance. For the n-th utterance of a customer-helpdesk dialogue of m utterances in total, we append (n/m) as the utterance position information. Both the Chinese and English corpora are preprocessed by using the same procedure as depicted above.

Secondly, we employ pre-trained Transformer networks to extract the hidden representations of the structured dialogue input. We explored the HuggingFace[32] and experimented with a variety of the Transformer models, including BERT, Chinese-BERT, Roberta, and their fine-tuned models, for obtaining the proper hidden representations of dialogue utterances. We put the details of the model selection and experimental comparison in section 4.

Topic information is also extracted as a part of the hidden representation of the input dialogue. Specifically, we employ a Latent Dirichlet Allocation (LDA) model to obtain the topic information from the input dialogue. The setting of topic number K has a crucial influence on the result of prediction with different text lengths, which is discussed in sec-

tion 5. The purpose of LDA is to infer the hidden topic structure using observable words which follows the process described in section 3.2. The topic probability vector of each dialogue is used as the input of the downstream module, together with the hidden representations extracted by the Transformer models. We use Jieba[2] for word segmentation and remove common stop_words to obtain more effective topic information.

Thirdly, the feedforward network is located above the dotted line, as shown in Fig. 3.6. The architecture of the feedforward network is arranged as follows: a full connection layer, an activation function, a dropout layer, a linear dimension reduction layer, and a softmax function. The full connection layer can be regarded as a weight layer, whose input and output dimensions are the same. Due to the role of the full connection layer, the model can act on both Chinese and English datasets and only need to fine-tune the parameters of the feedforward network, with the parameters of the extractor untouched. The linear layer takes the input of a 1546 to 1586 dimensional vector, and the output is a 5-dimensional vector, which is the probability distribution of the dialogue quality scores. The model has three linear layers, which output the A-score, S-score, and E-score respectively.

Finnally, the network obtains the probabilities over the quality label set $\Gamma = \{-2,-1,0,1,2\}$ for every quality type. More illustrations of the quality labels can be found in [83]. We employ the mean squared error (MSE) loss for evaluating the training loss. The model generates three distributions $\hat{y}_i^A$, $\hat{y}_i^S$, and $\hat{y}_i^E$ as the predictions for the A-score, S-score, and E-score of dialogue quality, respectively. We take the means of y_i over l human annotators for the A, S, and E scores as the targets, which are denoted by $\bar{y}_i^A$, $\bar{y}_i^S$, and $\bar{y}_i^E$ respectively. The training loss based on mean squared error is then given by:

$$\text{loss}(\bar{y},\hat{y}) = \sum_{\kappa\in\{A,S,E\}} \frac{1}{n}\sum_{i=1}^{n} (\bar{y}_i^{\kappa} - \hat{y}_i^{\kappa})^2 . \tag{3.1}$$

[2]https://github.com/fxsjy/jieba

In the ND subtask, we examine the proposed method on another set of labels, where each utterance is annotated whether it is a nugget. A nugget is a key turn that helps the customer transition from the current state (where the problem is yet to be solved) towards the target state (where the problem has been solved). These nuggets can effectively help researchers track the transition of customers' emotional states.

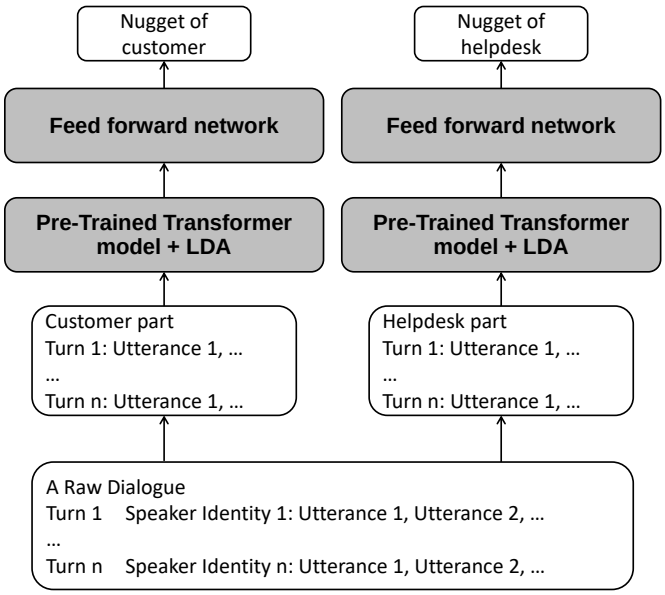

Fig. 3.7. The structure of nugget detection network.

We utilize the same dataset used in the dialogue quality subtask, but with different annotations. In order to fit the nugget labels, we applied a Sentence Rearrangement strategy, which divide all the dialogue utterances into two parts, namely the customer part and the helpdesk part. All modules in the dialogue quality subtask are carried over to the nugget detection subtask. We feed the concatenated utterances to the pre-trained Transformer models and the LDA topic model to obtain the Chinese and English hidden representations and employ a feedforward network to get the probability distribution of each nugget label.

The mean squared error loss is also used for training in nugget detection subtask. Detailed formula description can refer to Equation 3.1. A simple schematic diagram of

the nugget detection network is shown in Fig. 3.7.

3.3 Symbolic Logic Integration

In this section, we take Multi-class Multi-label Emotion Prediction as an example and give the corresponding grouding based on the LTN definition given in section 2.3. Suppose that one wants to learn a multi-label classifier P for the multi-class sentiment classification of a set of Weibo articles. Suppose that a set of training examples with multi-label emotions, including "Joy", "Hate", "Love", "Sorrow", "Anxiety", "Surprise", "Anger", and "Expect", is given on average. LTN uses the following language and grounding:

Domains:

articles, denoting the Weibo articles;

labels, denoting the emotion class labels.

Variables:

$x_{joy}, x_{hate}, x_{love}, x_{sorrow}, x_{anxiety}, x_{surprise}, x_{anger}, x_{expect}$ for the positive examples of each class;

x used to denote all the examples;

$D(x) = D(x_{joy}) = D(x_{hate}) = \cdots = articles$, where $D(.)$ is a function which returns the domain of a non-logical symbol given in input.

Constants:

$l_{joy}, l_{hate}, l_{love}, l_{sorrow}, l_{anxiety}, l_{surprise}, l_{anger}, l_{expect}$: the labels of each class;

$D(l_{joy}) = D(l_{hate}) = \cdots = labels.$

Predicates:

$P(x,l)$ denoting the fact that article x is labelled as l;

$D_{in}(P) = articles, labels$, where $D_{in}(.)$ is a function which returns the domain of the input of a function or predicate given in input.

Axioms:

$\forall x_{joy} P(x_{joy}, l_{joy})$: all the articles that include the emotion of "Joy" should have label l_{joy};

$\forall x_{hate} P(x_{hate}, l_{hate})$: all the articles include the emotion of "Hate" should have label l_{hate};

...: Same for "Love", "Sorrow", "Anxiety", "Surprise", "Anger", and "Expect";

$\forall x \neg(P(x, l_{joy}) \wedge P(x, l_{hate}))$: if an articles x is labelled as "Joy", it cannot be labelled as "Hate" too;

$\forall x \neg(P(x, l_{love}) \wedge P(x, l_{sorrow}))$: if an articles x is labelled as "Sorrow", it cannot be labelled as "Hate" too;

$\forall x \neg(P(x, l_{anxiety}) \wedge P(x, l_{surprise}))$: if an articles x is labelled as "Anxiety", it cannot be labelled as "Surprise" too;

$\forall x \neg(P(x, l_{anger}) \wedge P(x, l_{expect}))$: if an articles x is labelled as "Expect", it cannot be labelled as "Hate" too;

$\exists x(P(x, l_{joy}) \wedge P(x, l_{love}))$: if an articles x is labelled as "Joy", it maybe labelled as "Love" too;

$\exists x(P(x, l_{hate}) \wedge P(x, l_{sorrow}))$: if an articles x is labelled as "Hate", it maybe labelled as "Sorrow" too;

$\exists x(P(x, l_{anxiety}) \wedge P(x, l_{anger}))$: if an articles x is labelled as "Anxiety", it maybe labelled as "Anger" too;

$\exists x(P(x, l_{surprise}) \wedge P(x, l_{expect}))$: if an articles x is labelled as "Surprise", it maybe labelled as "Expect" too;

...: other similar emotional logics can also be defined with corresponding axioms.

Grounding:

$G(articles) = \mathbb{R}^{768}$: the articles are represented by a 768-dimensional hidden representations extracted by language models like BERT;

$G(labels) = \mathbb{N}^8$: we use an one-hot encoding to represent labels;

$G(x_{joy}) \subseteq \mathbb{R}^{m_1 \times 768}$, $G(x_{hate}) \subseteq \mathbb{R}^{m_2 \times 768}$, ... These sequences are not mutually-exclusive, one article can for instance be in both x_{joy} and x_{love}.;

$G(x) \subseteq \mathbb{R}^{m \times 768}$, that is, $G(x)$ is a sequence of all the articles;

$G(l_{joy}) = [1,0,0,0,0,0,0,0], G(l_{hate}) = [0,1,0,0,0,0,0,0], G(l_{love}) = [0,0,1,0,0,0,0,0]$, $\ldots$, $G(l_{expect}) = [0,0,0,0,0,0,0,1]$;

$G(P \mid \theta) : x, l \mapsto l^{\top} \cdot \sigma(MLP_{\theta}(x))$, where MLP has 8 output neurons corresponding to as many labels, and $\cdot$ denotes the dot product as a way of selecting an output for $G(P \mid \theta)$. In fact, multiplying the output by the one-hot vector $l^{\top}$ gives the probability corresponding to the label denoted by l. By contrast with the previous example in section 2.3, notice the use of a sigmoid function instead of a softmax function, for the labels are not mutually exclusive anymore.

In order to define our knowledge base (axioms), we need to define predicate P, constants, connectives, universal quantifier, and the SatAgg operator. For the connectives and quantifier, we use the stable product configuration. For predicate P, we use bert-base-chinese in this example. The constants represent the one-hot labels for the 8 classes, have already seen in the definition of the grounding above. Finally, let us define D the data set of all examples. The objective function is given by Formula 2.2 in section 2.3, where B is an mini batch from D.

3.4 Large Language Model Supervision

In this section, we will take Fine-Grained Argument Analysis (FinArg) tasks as examples to show the power of Large Language Model Supervision. The task of Fine-Grained Argument Analysis often encompasses various subtasks, each with distinct input and output in terms of both content and structure. To address this, we employ the T5 [36] model as the central backbone of our framework. T5 standardizes multiple NLP tasks into a unified text-to-text format, ensuring that both input and output are consistently represented as text strings. A comprehensive visualization of the Large Language Model Supervision framework is provided in Fig. 3.8.

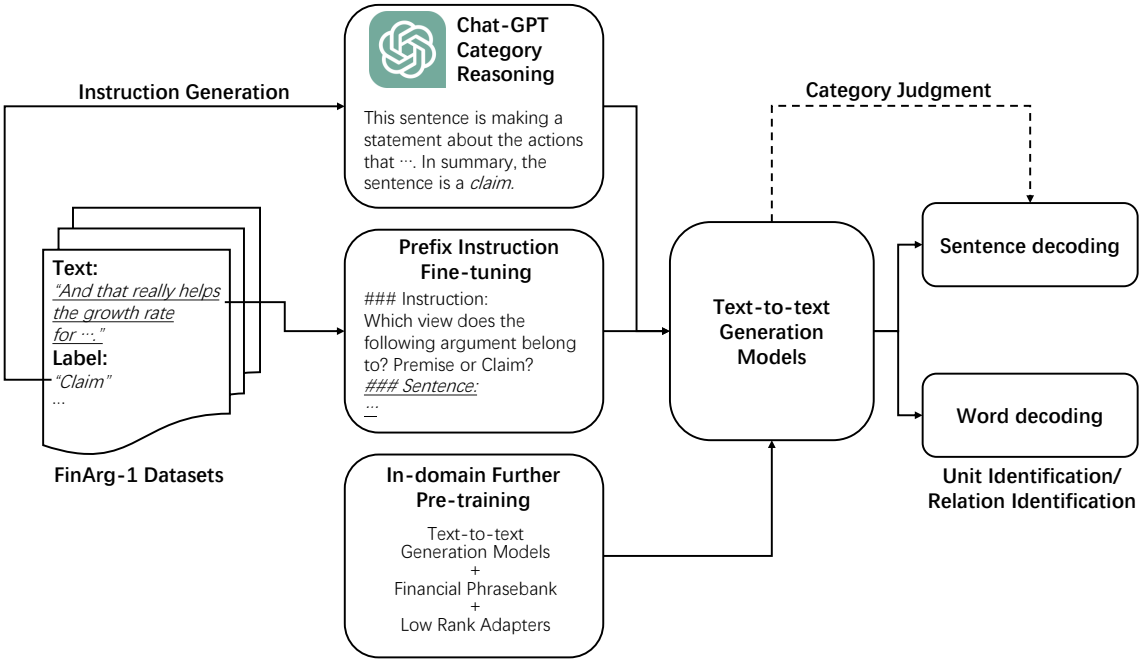

Fig. 3.8. Large language model supervision approach overview.

3.4.1 Prefix Instruction Fine-tuning

Both Prompt-based Learning [84] and Instruction Fine-tuning[85, 86] have been demonstrated to effectively enhance the performance of various LLMs. In the paradigm of prompt-based learning, the description of the task is embedded in the input. For example, instead of implicitly giving certain parameters to the model, they are directly input in the form of questions. To address different FinArg subtasks, we extensively tested a myriad of potential prompts and instructions. From this, we curated a generic prefix. For each subtask, we keep one long instruction and one short instruction respectively. Concurrently, we aimed to align the instructions utilized for fine-tuning the T5 model with the prompts adopted by ChatGPT. The specifics of these prompts can be found in Table 3.1. For an in-depth examination of how varying instructions influence the outcomes across different subtasks, please refer to Section 4. Subsequently, these refined datasets were employed to fine-tune the T5 model, as depicted in Fig. 3.8.

Tab. 3.1. Prompts and insturctions for different subtasks. ARI stands for Argument Relation Identification and AUI stands for Argument Unit Identification. {text} and {label} stand for the original dataset inputs and outputs respectively.

Sub tasks	Prompt/Instruction
Short Instruction for ARI	Judge the relationship between the two sentences. Attack/Support/None: {text_1}{text_2}
Long Instruction for ARI	Below are two sentences that contain opinions. Please judge the logical relationship between sentence 1 and sentence 2. The relationship can only be among Attack, Support, or no-relation. ### Sentence 1: {text_1} ### Sentence 2: {text_2}
Short Instruction for AUI	Premise or Claim:
Long Instruction for AUI	### Instruction: Which view does the following argument belong to? Premise or Claim? ### Sentence: {text} ### Argument:
Long Instruction for AUI (train with ChatGPT reasoning)	Below is a sentence belonging to an argumentation, contained with its component category of 'Premise' or 'Claim'. Write an explanation that appropriately explains which category the sentence belongs to and why the sentence falls into this category. Your explanation must end with 'In summary, the sentence is a premise' or 'In summary, the sentence is a claim'. ### Sentence: {text} ### Explanation:
ChatGPT category inference prompt	Below is a sentence belonging to an argumentation, **Paired** with its component category of 'Premise' or 'Claim'. Write an explanation that appropriately explains which category the sentence belongs to and why the sentence falls into this category. Your explanation must end with 'In summary, the sentence is a premise' or 'In summary, the sentence is a claim'. ### Sentence: {text} **### Category:** **{Label}** ### Explanation:

3.4.2 Within-domain Further Pre-training

Pre-training language models on specific in-domain data, known as domain-adaptive pre-training (DAPT), or on data relevant to particular tasks, termed task-adaptive pre-

training (TAPT), has been demonstrated to enhance performance in downstream tasks [87]. The nature of financial data is particularly apt for this approach. It is highly dynamic, deeply specialized, and has clear data demarcations. This makes it a prime candidate for further pre-training. Thanks to the commendable efforts of the huggingface team[3], fine-tuning and further pre-training are very convenient today.

In this paper, in addition to the original T5 model, we also fine-tune the Flan-t5[4] model through the financial phrasebank dataset[5] with Low-Rank Adapters (LoRA) [88]. The experimental results show that in some cases, further pre-training can indeed improve the effect of the model, but the instruction used in downstream tasks needs to be adjusted accordingly. See Section 4 for a detailed discussion and parameter settings.

3.4.3 Task-specific Decoding

Given that the T5 model always utilizes text strings for both input and output, we transformed the labels from the FinArg-1 task dataset into corresponding textual representations. Notably, when employing the "Explanation" generated by ChatGPT as the ground truth label, we consider the penultimate token of the output string as the model's label output word. This approach is adopted primarily because, in most instances, the last token of the string tends to be a period.

In the AUI subtask, we use the mapping $f_u : 0 \rightarrow$ 'premise', $1 \rightarrow$ 'claim' to map the corresponding labels. It is worth noting that words with a capitalized first letter cannot be used, because in the tokenizer of T5, words with capital letters are sometimes split into two tokens. In the ARI subtask, we use the mapping $f_r : 0 \rightarrow$ 'none', $1 \rightarrow$ 'support', $2 \rightarrow$ 'attack' to map the corresponding labels. When using "no relation" or "unrelated" as the mapping word, the tokenizer will also split them into 2 tokens. Finally, we choose "none" as the mapping word for label "0". Experimental results show that after at least one epoch of training, the output of the T5 model is always included in the mapping

[3]https://huggingface.co/

[4]https://huggingface.co/google/flan-t5-base

[5]https://huggingface.co/datasets/financial_phrasebank

vocabulary. Therefore, the final output label can be obtained without any other decoding.

When using the "Explanation" generated by ChatGPT as the ground truth label, due to the existence of instruction, almost all the text sequences output by the T5 model will end with words in the mapping vocabulary, just like the examples in Table 3.2. In some extreme cases, when the output of the T5 model exceeds the set max length parameter, the output will be truncated. This situation can usually be solved by limiting the output length of ChatGPT or increasing the max length of the T5 model. But for the sake of saving computing resources, we usually prefer a smaller max length. Therefore, when the output string does not end with the words in the mapping vocabulary, there are two ways to deal with it: 1) Simply considering the output to be wrong or, 2) Reinputing the output sentence into the fine-tuned model and judging what the category label contained in the sentence is, which we called Category Judgment. The results of the experiments are elaborated in section 4.

3.4.4 Causal Inference from ChatGPT

Inspired by Chain-of-Thought Prompting [89] and Causal Reasoning [90], we tried to equip the T5 model with the capability to not just classify the category of AUI, but also furnish explanations for its decisions, aiming to bolster classification accuracy. Initially, we employed a specific prompt, as shown in Table 3.1, to let the ChatGPT-3.5 model (model number 'text-davinci-003') generate an "Explanation" that rationalizes the classification determinations made. To ensure that the "Explanation" yielded by ChatGPT aligns with the label of the original dataset, we embedded a strong correlation constraint within the prompt (the bold part in Table 3.1). Under the restriction of strong constraints, all "Explanation" generated by ChatGPT conform to the ground truth label of the data. Table 3.2 contains some examples of explanations generated by ChatGPT.

Subsequently, the "Explanation" generated by ChatGPT is harnessed as the target text for the fine-tuning process of the T5 model, as shown in Fig. 3.8. When the reasoning of ChatGPT is used as the label, we aim to align the instruction during T5's fine-tuning

Tab. 3.2. "Explanation" examples generated by ChatGPT.

Label	ChatGPT "Explanation"
Premise	"This sentence provides evidence to support a larger argument by providing an example of how success can be achieved. It does not make a statement that can be proven or disproven, but rather provides an example of how success can be achieved. In summary, the sentence is a premise."
Claim	"This sentence is making a statement about the actions that Amazon is taking, which is a claim. It is not providing any evidence or reasoning to support a conclusion, which is what a premise would do. In summary, the sentence is a claim."

with the prompt that was initially utilized by ChatGPT, as shown in "Long Instruction for AUI (train with ChatGPT reasoning)" in Table 3.1.

3.5 Conclusion

In conclusion, the third section has comprehensively detailed the workflow of the proposed Interpretable Text Sentiment Analysis Framework. This innovative framework is strategically divided into three main components, each playing a crucial role in enhancing the effectiveness and interpretability of text sentiment analysis.

Firstly, the Enriched Semantic Feature Fusion part focuses on extracting and amalgamating diverse semantic features from the text. This process not only captures the explicit sentiments expressed but also uncovers the topic information, ensuring a more holistic understanding of the text's sentiment.

Secondly, the Symbolic Logic Integration introduces a layer of logical reasoning to the framework. By embedding symbolic logic, this component brings a high degree of interpretability and reliability to the analysis. It allows the framework to process textual data in a manner that is not only analytically profound but also aligns closely with human logical reasoning.

Lastly, the Large Language Model Supervision employs advanced language models to oversee and refine the analysis process. This supervision ensures that the framework

stays updated with the latest linguistic trends and nuances, thereby enhancing its ability to reason, infer causality, and provide explanations.

Together, these three components form a robust, interpretable, and dynamic framework for text sentiment analysis. This framework not only addresses the current challenges in sentiment analysis but also sets a new standard for future developments in this field, promising enhanced accuracy and deeper insights into the vast and varied terrain of human emotions as expressed through text.

4 Experiments and Analysis

In the current study, our proposed framework is evaluated on four datasets in five subtasks for text sentiment analysis, which details are as follows.

4.1 Setup

4.1.1 Datasets

1. **DCH-2**: Dialogues contains 4,390 real customer-helpdesk conversation in Chinese and their English translations. It contains 2 subtasks: **Dialogue quality prediction (DQ)**, which assign quality scores to each dialogue in terms of three subjective criteria: task accomplishment, customer satisfaction, and dialogue effectiveness; **Nugget detection (ND)**, which classify whether a customer or helpdesk turn is a nugget, where being a nugget means that the turn helps towards problem solving. The statistics of the above-mentioned datasets are described in more details in Table 4.1

Tab. 4.1. DCH-2 statistics. Each dialogue was annotated by 19 or 20 annotators independently.

(a) Total number and ratio of dialogue quality labels over all 4,390 dialogues

	-2	-1	0	1	2
Task accomplishment	13937 (16.6%)	15497 (18.5%)	33810 (40.4%)	13659 (16.3%)	6807 (8.1%)
Customer satisfaction	12877 (15.4%)	14829 (17.7%)	36754 (43.9%)	13334 (15.9%)	5916 (7.1%)
Dialogue effectiveness	12643 (15.1%)	12308 (14.7%)	24810 (29.6%)	25397 (30.3%)	8552 (10.2%)

(b) Total number and ratio of turn-level nugget type labels over all 4,390 dialogues

	Trigger	Regular	Goal	Not_a_Nugget
Customer turns	71925 (37.0%)	71115 (36.6%)	8079 (4.2%)	43186 (22.2%)
Helpdesk turns	N/A	93542 (59.4%)	23557 (15.0%)	40341 (25.6%)

2. **Ren_CECps**: Weibo articles consists of 1,486 documents with 36,525 sentences. Based on this corpus, we perform the **Multi-label emotion prediction (MEP)** task, which predict multi-label emotion and sentiment intensity contained in text at sentence, paragraph, and document level. The statistics of the above-mentioned datasets are described in more details in Table 4.2.

Tab. 4.2. The emotional distribution of multi-label sentences in Ren_CECps over all 36525 sentences with eight emotional categories (joy, hate, love, sorrow, anxiety, surprise, anger, expect).

Laber amount	Sentence amount	per.(%)
no-emo	2753	7.54
1	19998	54.75
2	11731	32.12
3	1847	5.06
4	175	0.48
5	15	0.04
6	6	0.02
total	36525	100

3. **Argument Unit Identification (AUI):** This subtask requires models to distinguish whether the given argumentation sentence as a claim or a premise. The data set has a total of 9,691 sentences, of which 5,078 are premises and 4,613 are claims.

4. **Argument Relation Identification (ARI):** This subtask necessitates the identification of the relationship, specifically discerning whether it's one of support, attack, or other. The text portion of the dataset consists of two separate sentences. It is worth mentioning that this dataset is a very imbalanced dataset. The "attack" label accounted for only 1.1% Among the total label. Table 4.3 show the statistics of the datasets.

Tab. 4.3. Data statistics of argument unit identification and argument relation identification.

Argument Unit Identification				
	Train	Dev	Test	Whole
Preminse	4,062	508	508	5,078(52.4%)
Claim	3,691	461	461	4613(47.6%)
Total	7,753	969	969	9,691
Argument Relation Identification				
	Train	Dev	Test	Whole
Support	3,859	482	482	4,823(69.9%)
Attack	62	8	8	78(1.1%)
Other	1,600	200	200	2,000(29.0%)
Total	5,521	690	690	6,901

4.1.2 Computer Configuration

All experiments were run on the following servers. OS: CentOS Linux release 7.6.1810. Linux Core: 3.10.0-957.el7.x86 64. CPU: Intel Core i7 6700k. GPU: NVIDIA GeForce RTX 3090Ti 24 GB. RAM: TeamGroup 32 GB. Python version: 3.7.3.

4.1.3 Evaluation metrics

Cross-bin metrics are preferable to bin-by-bin metrics because the classes of DQ subtasks are non-nominal. As a result, we employ two cross-bin metrics: Normalised Match Distance (NMD) and Root Symmetric Normalised Order-aware Divergence (RSNOD) in DQ subtask[91]. Unlike the DQ subtask, the classes in the ND subtask are nominal, so bin-by-bin metrics are more appropriate. In particular, the ND subtask employs two metrics: Root Normalised Sum of Squares (RNSS) and Jensen-Shannon Divergence (JSD)[92]. In the Ren_CECps corpus, Multi-label emotion prediction task is more similar to the multi-label classification task. To better compare with similar works, we chose the Accuracy, Precision, Recall, F1-score, One Error, and Average Precision as our metrics, which are widely used in text classification and sentiment analysis tasks[93][94]. In the ARI and AUI tasks, we report both Micro-F1 and Macro-F1 scores. Due to the presence of imbalanced datasets, we use Macro-F1 to rank the results. These metrics are calculated as follows:

$$\mathrm{NMD}\,(p, p^*) = \frac{\mathrm{MD}\,(p, p^*)}{L-1}. \tag{4.1}$$

$$\mathrm{RSNOD}\,(p, p^*) = \sqrt{\frac{\mathrm{SOD}\,(p, p^*)}{L-1}}, \tag{4.2}$$

where MD presents for the sum of absolute errors compared from the cumulative probability distributions and SOD presents for Symmetric Order-Aware Divergence, respec-

tively.

$$\text{RNSS} = \sqrt{\frac{\sum_{i \in A} \left(p(i) - p^*(i)\right)^2}{2}}. \tag{4.3}$$

$$\text{JSD}\left(p \| p^*\right) = \frac{\text{KLD}\left(p \| p_M\right) + \text{KLD}\left(p_M \| p^*\right)}{2}, \tag{4.4}$$

$$\text{where KLD}\left(p_1 \| p_2\right) = \sum_{\text{is.t. } p_1(i)>0} p_1(i) \log_2 \frac{p_1(i)}{p_2(i)}. \tag{4.5}$$

$$\text{Precision} = \frac{TP}{TP+FP}, \tag{4.6}$$

$$\text{Recall} = \frac{TP}{TP+FN}, \tag{4.7}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{4.8}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}, \tag{4.9}$$

where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively[93]. Specifically, Micro-F1 calculates metrics globally by counting the total true positives, false negatives and false positives. Macro-F1 calculates metrics for each label, and find their unweighted mean.

$$\text{OneError} = \frac{1}{n} \sum_{i=1}^{n} \delta \left[\underset{\mathbf{e_t}}{\text{argmax}} g_t\left(x_i\right) \notin R_i \right], \tag{4.10}$$

where One Error (OE) evaluates the fraction of sentences whose top-ranked emotion is not in the relevant emotion set.

$$\begin{aligned} &\text{AveragePrecision} = \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \tfrac{1}{|R_i|} \times \\ &\left(\textstyle\sum_{t:e_t \in R_i} \left|\left\{e_s \in R_i \mid g_s\left(x_i\right) > g_t\left(x_i\right)\right\}\right|\right) / \left(\left|\left\{e_s \mid g_s\left(x_i\right) > g_t\left(x_i\right)\right\}\right|\right), \end{aligned} \tag{4.11}$$

where Average Precision(AP) evaluates the average fraction of the relevant emotions ranked higher than a particular emotion.

4.2 Customer Service Dialogue Evaluation

In this section, comparative experiment results are provided. In DQ subtask, our proposed method is compared with 7 runs for Chinese dialogue and 6 runs for English dialogue respectively. In ND subtask, 8 runs for both Chinese and English dialogues are considered.

4.2.1 Dialogue Quality Prediction

In this section, we report the brief results of three types of DQ prediction scores for both Chinese and English subtasks in Table 4.4 and 4.5. The specifics of each run are detailed as follows:

1. Chinese_TUA1_run0: similar to the method from TUA1 team proposed in NTCIR 15 DialEval-1 Task[95], simply replace BERT with Roberta.

2. Chinese_TUA1_run1: proposed dialogue quality prediction network in section 3.2 with topic nunber of 20 and dropout probability of 0.2.

3. Chinese_TUA1_run2: proposed dialogue quality prediction network with topic nunber of 10 and dropout probability of 0.1.

4. English_TUA1_run0: same model structure as Chinese_TUA1_run2, just retrain the parameters with English dev dataset. Note that although the best results for the Chinese and English subtasks of DQ in the overview paper are run_2 and run_0 respectively, they are actually the same model structure with the same hyperparameters.

Among all types of scores, the model of run_2 in Chinese subtask and run_0 in English subtask achieved the highest overall score. Note that they are the same model. The model structure can refer to Fig. 3.6 with a topic number of 10 and dropout probability of 0.1, thus the dimension of the concatenated hidden representation is 1546. We employ

Tab. 4.4. Results for Chinese dialogue quality prediction. “TUA1” is our team name. The bold font in the table indicates the best result in DialEval-2 task. “↓” indicates “the smaller the better”. A-score, S-score, E-score present task accomplishment, customer satisfaction, and dialogue effectiveness, respectively.

Score type	Run	Mean RSNOD↓	Mean NMD↓
A-score	TUA1-run2	**0.1992**	**0.1325**
	TUA1-run1	0.2092	0.1369
	TUA1-run0	0.2154	0.1474
	Baseline-run0	0.2301	0.1772
	Baseline-run2	0.2320	0.1577
	RSLDE-run0	0.2438	0.1537
	RSLDE-run1	0.2446	0.1551
	IMNTPU-run0	0.2479	0.1618
	Baseline-run1	0.2767	0.2500
	NKUST-run0	0.2774	0.2453
S-score	TUA1-run2	**0.1758**	0.1166
	TUA1-run1	0.1840	**0.1159**
	TUA1-run0	0.1884	0.1305
	RSLDE-run0	0.1938	0.1243
	RSLDE-run1	0.1964	0.1229
	Baseline-run0	0.1998	0.1523
	IMNTPU-run0	0.2032	0.1315
	Baseline-run2	0.2062	0.1288
	NKUST-run0	0.2732	0.2293
	Baseline-run1	0.2959	0.2565
E-score	TUA1-run0	**0.1545**	**0.1136**
	TUA1-run1	0.1647	0.1262
	RSLDE-run0	0.1660	0.1222
	TUA1-run2	0.1671	0.1310
	RSLDE-run1	0.1725	0.1286
	Baseline-run0	0.1854	0.1579
	IMNTPU-run0	0.1860	0.1427
	NKUST-run0	0.2253	0.1897
	Baseline-run1	0.2496	0.2106
	Baseline-run2	0.2569	0.1710

"roberta-base-finetuned-jd" as the Chinese Pre-trained Transformer model and "roberta-base" as the English Pre-trained Transformer model. The last 2 layer parameters of the model are unfrozen for fine-tuning. A detailed discussion of Transformer model selection and unfreeze layers can be found in discussions section.

Tab. 4.5. Results for English dialogue quality prediction. "TUA1" is our team name. The bold font in the table indicates the best result in DialEval-2 task. "↓" indicates "the smaller the better". A-score, S-score, E-score present task accomplishment, customer satisfaction, and dialogue effectiveness, respectively.

Score type	Run	Mean RSNOD↓	Mean NMD↓
A-score	TUA1-run0	**0.1967**	**0.1327**
	Baseline-run2	0.2320	0.1577
	Baseline-run0	0.2321	0.1780
	IMNTPU-run0	0.2535	0.1654
	RSLDE-run0	0.2615	0.1957
	RSLDE-run1	0.2725	0.1896
	Baseline-run1	0.2767	0.2500
S-score	TUA1-run0	**0.1855**	**0.1214**
	Baseline-run0	0.1986	0.1467
	IMNTPU-run0	0.2020	0.1312
	Baseline-run2	0.2062	0.1288
	RSLDE-run0	0.2078	0.1381
	RSLDE-run1	0.2154	0.1438
	Baseline-run1	0.2959	0.2565
E-score	TUA1-run0	**0.1742**	**0.1360**
	Baseline-run0	0.1745	0.1431
	IMNTPU-run0	0.1826	0.1400
	RSLDE-run0	0.1832	0.1429
	RSLDE-run1	0.1889	0.1444
	Baseline-run1	0.2496	0.2106
	Baseline-run2	0.2569	0.1710

4.2.2 Nugget Detection

In this section, we report the result of nugget detection in Table 4.6, which is evaluated based on the mean JSD metric and the mean RNSS metric, respectively. The specifics of each run are detailed as follows:

1. Chinese_TUA1_run0: proposed nugget detection network in section 3.2.
2. Chinese_TUA1_run1: similar to the method from TUA1 team proposed in NTCIR 15 DialEval-1 Task, simply replace BERT with Roberta.
3. English_TUA1_run0: same model as Chinese_TUA1_run0.

Tab. 4.6. Results for nugget detection. "TUA1" is our team name. The bold font in the table indicates the best result in DialEval-2 task. "↓" indicates "the smaller the better".

Language	Run	Mean JSD↓	Mean RNSS↓
Chinese	RSLDE-run0	**0.0560**	**0.1604**
	Baseline-run0	0.0585	0.1651
	RSLDE-run2	0.0607	0.1720
	RSLDE-run1	0.0634	0.1712
	NKUST-run0	0.0670	0.1761
	TUA1-run0	0.0700	0.1780
	Baseline-run2	0.1864	0.2901
	Baseline-run1	0.2042	0.3371
	NKUST-run1	0.2432	0.3774
English	RSLDE-run0	**0.0557**	0.1615
	IMNTPU-run0	0.0601	**0.1574**
	Baseline-run0	0.0625	0.1722
	NKUST-run0	0.0641	0.1744
	RSLDE-run2	0.0676	0.1778
	RSLDE-run1	0.0691	0.1853
	TUA1-run0	0.0728	0.1830
	Baseline-run2	0.1864	0.2901
	Baseline-run1	0.2042	0.3371

The experimental results show that our proposed model, although not the best among the participants, still exceeds the baseline using uniform and popularity prediction. For other teams and the method used by baseline, please refer to the official paper of the DialEval-2 task[96].

4.2.3 Selection of Pre-trained Transformer Model

One of the reasons that transformer-based models have received so much focus is that they can be quickly fine-tuned for diverse tasks and domains without considerable pre-training. Tansformer-based models trained on different pre-datasets have varied parameters and obtained text features, as well as different application effects in multiple domains. In this section, we discuss the influence of different Transformer models in DQ subtask. Thanks to the Hugging Face model library [6], we can easily try a variety

[6]https://huggingface.co/models

of pre-trained models. Several of the Chinese and English model pairs we tried and their experimental results are listed in Table 4.7. Notation "-" represents English dialogue datasets not used in the model. "Score-sum" denotes the summation of RSNOD and NMD scores for all A, S, and E scores. The distance scores were transformed by $-log()$ for readability. Thus, the higher the transformed scores, the better the model's effectiveness.

Tab. 4.7. Results for different pre-trained transformer models. "↑" indicates "the larger the better" and "-" indicates not used in model.

Chinese_model	English_model	Score_sum↑
bert-base-chinese	-	13.37
chinese-bert	-	14.59
bert-base-chinese	bert-base-cased	14.56
chinese-roberta-base	roberta-base	16.14
roberta-base-finetuned-jd	roberta-base	**16.27**

The best result is achieved by "roberta-base-finetuned-jd"[7] for Chinese dialogue and "roberta-base" for English dialogue. "roberta-base-finetuned-jd" is a Chinese RoBERTa-Base model fine-tuned with the "JD full"[97] dataset, which consist of user reviews of different sentiment polarities. The experimental results show that using a pre-trained model in which data type and domain are closer to a certain task, or making appropriate domain adaption, can significantly improve the accuracy of model predictions.

We also experimented with several methods for unfreezing the parameter layers in Transformer model training. All models used in the tests are based on the original BERT_BASE model, which has 12 layers of bidirectional Transformer blocks and one layer of pooler. The experimental results reveal that having too many or too few unfreeze layers reduces model prediction accuracy and may even invalidate the final prediction result. Table 4.8 shows the entire comparison test. When the last two Transformer layers and the pooling layer were unfrozen, the best results were achieved. It's worth noting that if we unfreeze all of the pre-trained model's parameters, the GPU will run out of memory

[7] https://huggingface.co/uer/roberta-base-finetuned-jd-full-chinese

and training will be disabled.

Tab. 4.8. Results for different unfreeze layers. "↑" indicates "the larger the better" and "-" indicates GPU out of memory.

unfreeze layers	Score_sum↑
none	13.37
pooler only	14.14
pooler & layer.11	16.14
pooler & layer.10-11	**16.27**
pooler & layer.9-11	15.96
pooler & layer.8-11	13.86
pooler & layer.6-11	13.77
all	-

Compared with the massive quantity of data required for pre-training, the training method of unfreezing several parameter layers of Transformer and fine-tuning can be well adapted to training on small datasets. All training processes can converge in less than 20 epochs, with a suitable learning rate. Depending on the number of unfreezing parameter layers, each training epoch takes a different training time. The training time for each epoch on the server mentioned above is about 9 minutes at most and 1 minute at least.

4.2.4 Neuro or Symbolic

In this section, we attempt to answer the question "Neuro or Symbolic?" through some non-qualitative experiments. To evaluate whether neuro or symbol plays a greater role in the overall model, we compared the shapeley value of model input with the multi-head attention weight of BERT and the high-frequency topic words of LDA clustering. SHapley Additive exPlanations (SHAP) [98] is a model-agnostic method for interpreting the predictions of machine learning models. It assigns a feature importance score to each input feature, based on its contribution to the model's output. SHAP values are calculated using the Shapley values, which measure the contribution of each feature to the model's output on average, across all possible combinations of features. Shapley values can be

easily implemented by calling the SHAP library[8]. BertViz[9] is an interactive tool for visualizing attention in Transformer language models such as BERT. It can demonstrate the weight of attention across all layers and heads. The higher the attention weight, the more BERT pays attention to a certain token in a certain layer. Furthermore, we highlight the top 10 high-frequency topic words of each topic in LDA to represent the most concerned words of LDA, as shown in Fig. 4.4.

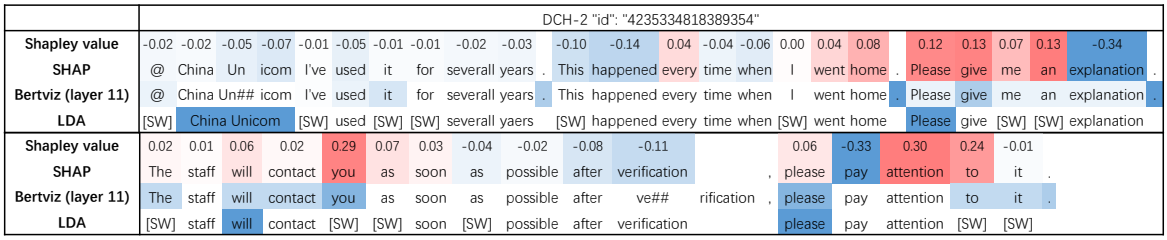

Fig. 4.1. Visualization of the impact of each module on the classification results.

We performed the analysis on samples from the DCH-2 dataset using the method described above. The results of the comparison are shown in Fig. 4.1. "[SW]" represents the stop words that are ignored when LDA builds the word dictionary. "##" represents one word divided into multiple tokens through BERT tokenizer. The special tokens [CLS] and [SEP] are omitted. It is worth noting that the darker the color of the block, the bigger the value of attention or shapley. In the shapley value, red means that this token shifts the result tend to be positive, and blue vice versa. In Bertviz and LDA, the color of the block does not represent positive or negative polarity, but only represents the attention weight or top-frequency topic words. The Shapley value is a verbatim interpretation of the entire model. Bertviz shows how the last hidden layer of the Transformer model pays attention to different tokens. LDA highlights the top 10 high-frequency topic words. The highlighted words of SHAP show that its contribution to emotional prediction is in line with human common sense. For example, the word "explanation" is more inclined to negative emotions in human cognition. Bertviz highlighted words contain more demon-

[8]https://github.com/slundberg/shap

[9]https://github.com/jessevig/bertviz

strative pronouns and punctuation marks, which shows that the Transformer model is more concerned with the grammatical structure and implication logic of the sentence (referring only to the last attention layer). LDA, on the other hand, focuses on the keywords or commonality of certain topics, such as "China Unicom", which are sometimes ignored by the Transformer model. In general, Bertviz shows how the Transformer model pays attention to different tokens and LDA focuses on high-frequency topic words that may be ignored by the Transformer model. The final SHAP value shows that the highlighted emotional words of the model output are in line with human common sense. Each module provides unique insights into the model's workings, and all are important for a comprehensive understanding.

4.2.5 Fast Cross-Task Training

In this section, we conduct some comparative experiments to verify the fast cross-task training ability of the proposed method. We employ the "roberta-base-finetuned-jd-binary-chinese"[10] as the pre-trained Transformer model and an LDA topic model with the topic number of 20. Then we randomly select a trainset of 5000 sentences from the "JD binary" dataset[11]. This trainset is used to train the feedforward network as shown in Fig. 3.3, with freezing all the parameters of "roberta-base-finetuned-jd-binary-chinese" model. Because the "JD binary" dataset is a binary polarity dataset, the number of labels finally output by the feedforward network is set to 2. After 5 epochs of training, the prediction accuracy tends to converge. The LDA topic model is also trained on this dataset.

Then we use Ren_CECps dataset to verify the cross-task learning ability of the proposed method. In addition to multi-label emotional labels, Ren_CECps has binary polarity labels as well. We retrain the LDA model to extract different topic information and then use the same Transformer model and feedforward model without fine-tuning as

[10]https://huggingface.co/uer/roberta-base-finetuned-jd-binary-chinese

[11]https://github.com/zhangxiangxiao/glyph

mentioned in the previous paragraph to predict the binary results. The LDA model is an unsupervised approach and requires quite limited training time (takes an average of 6.74 seconds across 4 experiments on the device described in Section 5.1.2).

Tab. 4.9. Results cross-task learning. "RBFJBC" presents for "roberta-base-finetuned-jd-binary-chinese".

model	dataset	accuracy↑
RBFJBC	JD binary	0.972
RBFJBC + LDA	JD binary	**0.977**
RBFJBC	Ren_CECps	0.793
RBFJBC + LDA	Ren_CECps	**0.836**

The experimental results are shown in Table 4.9. The results demonstrate that our proposed method can further improve the accuracy of the pre-trained model with almost negligible further training.

4.3 Multi-label Emotion Prediction and Intensity Analysis

4.3.1 Experimental Results

To further prove the effectiveness of our model in sentiment analysis of Chinese text, we compared our proposed model with the state-of-the-art research results, which are also based on the Ren_CECps corpus like [99, 94, 55, 100]. We also selected some classic machine learning algorithms and transformer-based networks as our baselines, like TF-IDF and BERT. The details of the baselines are as follows:

TF-IDF[99]: a basic Term Frequency–Inverse Document Frequency method.

TF-IDF_word2vec[99]: an enhanced TF-IDF method utilizes word embeddings learned by word2vec as weights for each associated word.

WMD[99]: an emotion-separated method to assign the emotion labels of sentences with different values utilizes the Word Mover' s Distance algorithm as a way of feature representation.

Rank-SVM[100]: adapts the maximum margin strategy to deal with multilabel data, focusing on distinguishing relevant from irrelevant labels while ignoring relevance rankings.

RER[94]: a novel framework based on Relevant Emotion Ranking to identify multiple emotions from emotion relationships with text constraint.

RERc[94]: extends RER by incorporating emotion relationships as constraints into the learning framework.

HNet[100]: a Hierarchical Network with label embedding for contextual emotion recognition.

DATN-1[55]: a Dual Attention Transfer Network which divides the sentence representation into two different feature spaces. DATN-1 obtains the shared attention weights of source text first.

DATN-2[55]: the same architecture as DATN-1, which computes the attention weights of target text first.

RoBERTa: we use "xlm-roberta-base"[12] as the baseline of pre-traind RoBERTa in this experiment.

BERT: we use "bert-base-chinese"[13] as the baseline of pre-traind BERT in this experiment.

We chose macro F1-score as our primary evaluation metric from among the numerous evaluation standards. One of the reasons is that all the papers we chose for comparison utilized the same F1-score measures. Other metrics, on the other hand, cannot appropriately evaluate the experimental results due to some specific characteristics of the corpus. In the Ren_CECps corpus, two or more sentiment labels appear in a significant portion of sentences at the same time, and the sentiment intensities are occasionally annotated with

[12]https://huggingface.co/xlm-roberta-base

[13]https://huggingface.co/bert-base-chinese

the same value. However, none of the comparison papers addressed how to assess the accuracy of the results in this scenario. In addition, we provide the other metrics mentioned in section 5.1.3 for reference.

Tab. 4.10. Compared with similar works on Ren_CECps, "↑" indicates "the larger the better", "↓" indicates "the smaller the better", and "-" indicates not provided in the original paper. The first four methods are based on traditional machine learning or its variants. The latter five methods are based on deep learning and attention mechanisms.

Model	Acc↑	Precision↑	Recall↑	macro F1-score↑	Average Precision↑	One-error↓
TF-IDF	-	0.203	0.190	0.197	-	-
TF-IDF_word2vec	-	0.239	0.231	0.235	-	-
WMD	-	0.338	0.300	0.318	-	-
Rank-SVM	-	-	-	0.397	0.583	0.561
RER	-	-	-	0.410	0.675	0.456
RERc	-	-	-	0.416	0.680	0.455
HNet	-	-	-	0.419	-	**0.356**
DATN-1	0.393	-	-	0.410	0.670	0.501
DATN-2	0.457	-	-	0.444	0.674	0.498
RoBERTa	**0.528**	0.428	0.449	0.429	0.669	0.410
BERT	0.516	0.427	0.472	0.431	0.682	0.379
Our work	0.525	**0.479**	**0.496**	**0.484**	**0.695**	0.363

The results demonstrate that our model can compute the emotional intensity of each emotion label at the same time. In the instance where a sentence includes two sentiment tags with the same intensity, we calculate the model classification accuracy with a 50% accuracy rate. In Table 4.10, we can observe that our model outperformed all emotion categorization techniques mentioned above on macro F1-scores.

4.3.2 What does the LDA learned?

The number K of LDA topics has a significant impact on the outcome of emotional computing. The use of a different number of topics will result in unequal classification effects. At the same time, the optimum value of K is incompletely homologous during the extraction of representations at different levels or in different tasks.

As shown in Fig. 4.2, is an example of different classification results with varying

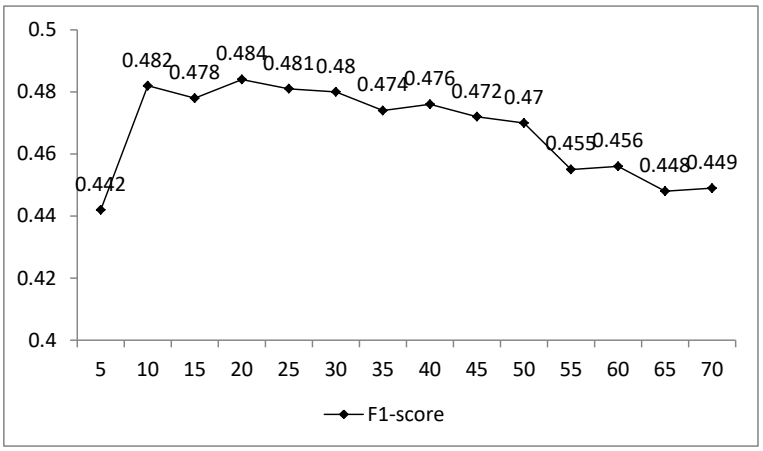

Fig. 4.2. Different results for different topic numbers in MEP task, with sentences level bert-base-chinese and paragraphs level LDA.

numbers of topic K in MER task. In general, it will get a higher F1-score with a K number between 10 and 30, and in this example, the F1-score reaches the highest value when K is 20. If K is too small, the distance between the extracted topic features is too short, and the probability distribution of different emotions under different topics cannot be distinguished well. If K is too large, the extracted topic features will instead become noise, which does not work on classification results.

We also verify the above-mentioned conclusions on the ND task. When the number of topics is 10, the model achieves the best prediction results, as shown in Fig. 4.3.

We performed some case studies to see what LDA learned during topic clustering. In LDA's topic clustering results on MEP task, the same topic words usually contain different emotions, as shown in Fig. 3.4. The probability of the topic words goes from left to right. This is an example of topic extraction, with a total topic number of 20 and clustering level of the paragraph. The gray blocks indicate the word "like", the white blocks with boxes indicate "woman", and the black blocks indicate "maybe", respectively. Text with both the topic words "like" and "women" will have a high probability of emotions containing "love" and "joy", however, text with topic words of "like" and "maybe" will have a high probability of "sorrow" and "anxiety". When we combine the

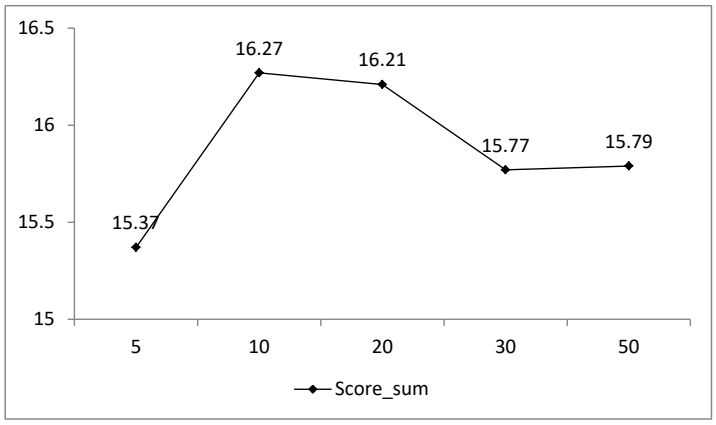

Fig. 4.3. Different results for different topic numbers in DQ task.

hidden representation extracted by BERT with the topic probability of LDA, the previous attention will be more inclined to “like” and “maybe”. This helps the model to recognize the sentiment of the sentence better.

It is worth noting that not all topic words are single English words (made of two Chinese words), but the high-frequency keywords in the sample happen to be single English words. Furthermore, concatenating different levels of topic extraction on hidden representations of sentences, paragraphs, or documents has a significant impact on the classification result.

Another case study for what LDA learned can be referred to Fig. 4.4 on DCH-2 dataset. When “China Unicom” (gray blocks, a Chinese telecom company) and “Signal” (the white block with boxe) appear in the same dialogue, the customer satisfaction and task accomplishment scores for dialogue quality always tend to be low because China Unicom’s signal is admittedly poor.

Compared to the “black box” of the DNN network, these case studies partly explain why our proposed method can accurately predict the sentiment contained in the text, within the range of human comprehension. The fully-connection layer of the feed-forward network is equivalent to a weight layer, which is used to determine the weight of the visible LDA topic features and the hidden representations extracted by the invisible

Topic	Topic Words				
1	Mobile phone	Hammer	Technology	Hello	Question
2	Mobile phone	Download	Vivo	Hello	Software
3	China Telecom	Broadband	Hello	Telecom	Private message
4	4G	Support	Thanks	Network	User
5	China Unicom	Hello	Service	Unicom	123456789
6	Youbao	Machine	Vending machine	Serial number	Drinks
7	China Unicom	Hello	Unicom	Signal	Question
8	Phone number	Question	Hello	Thanks	Solve
9	91	LeTV	Helper	Software	Mobile phone
10	China Unicom	Hello	Question	Thanks	Reply

Fig. 4.4. Topic clustering with the total topic number of 10 on DCH-2.

Transformer, which ultimately play a key role in predicting the outcome. The following section contains more information about the relationship between the topic information extracted by LDA and the final classification results. Furthermore, since LDA topic clustering is an unsupervised and fast process, it can facilitate cross-task learning. That will be discussed in the next section.

4.4 Financial Argument Analysis

4.4.1 Experimental Results

To prove the effectiveness of our proposed framework in fine-Grained argument understanding tasks within the domain of financial analysis, we compared our proposed model with some strong baselines. We also compared the results from the state-of-the-art LLMs, such as GPT-4. The details of the baselines are as follows:

BERT: [101] is an NLP model developed by Google's AI, which stands for Bidirectional Encoder Representations from Transformers. We use "bert-base-uncased"[14] as the baseline of pre-traind BERT in this experiment. The hidden representation of the [CLS] token is extracted, and a single-layer MLP is added for label classification.

[14]https://huggingface.co/bert-base-uncased

RoBERTa: [29] is an NLP model builds upon the BERT architecture, utilizing dynamic masking and larger batch size. We use "xlm-roberta-base"[15] as the baseline of pre-traind RoBERTa in this experiment. We added same MLP layer as BERT.

FinBert: [79] is a pre-trained NLP model to analyze sentiment of financial text. It is built by further training the BERT language model in the finance domain, using a large financial corpus.

T5: [36] is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks and for which each task is converted into a text-to-text format. We use "t5-large"[16] with 770 million parameters as baseline. The text and labels of the dataset are directly used as input and output of T5, without any prompts.

ChatGLM: [102] is an open bilingual language model based on General Language Model (GLM) framework. We choose ChatGLM-6B[17], with 6.2 billion parameters, as the baseline.

GPT-4: [35] is a large multimodal model accepting image and text inputs and emitting text outputs, which was recently released by OpenAI. Since the parameters of GPT-4 is not public, we use the few-shot method to complete the experiment.

TMUNLP: [103] uses a voting strategy to determine the optimal output from several language models.

IDEA: [104] adds a multi-layer convolution mechanism based on the text features extracted by BERT to improve the robustness of argument analysis.

LIPI: [105] uses pre-trained language models like BERT-SEC [106] and FinBERT, and a cross-encoder architecture to handle deep semantics and relationships.

For comparative experiments, we also use "t5-large" as the backbone of the proposed framework. BERT-like models use a learning rate of $5e^{-5}$ while other models use a

[15]https://huggingface.co/xlm-roberta-base
[16]https://huggingface.co/t5-large
[17]https://github.com/THUDM/ChatGLM-6B

learning rate of $3e^{-4}$. All experiments were trained for 10 epochs, and the final result was the best result among 5 runs.

Tab. 4.11. Experiment sesults for AUI and ARI subtasks. 'CIC' presents causal inference from ChatGPT and 'FFP' presents financial further pre-training. Bold fonts represent the best results.

	Argument Unit Iden.		**Argument Relation Iden.**	
	Micro-F1	**Macro-F1**	**Micro-F1**	**Macro-F1**
BERT	75.33	75.32	82.78	52.85
RoBERTa	74.93	74.91	81.92	55.67
FinBert	75.80	75.52	82.69	51.85
T5	73.89	73.82	82.19	54.03
ChatGLM	76.47	76.47	79.63	60.17
GPT-4 (few-shot)	62.51	62.49	69.88	49.00
TMUNLP	76.57	76.55	82.07	57.90
IDEA	76.47	76.46	81.74	51.85
LIPI	73.89	73.86	79.42	60.22
Ours	**77.40**	**77.32**	85.65	**61.50**
Ours w/o CIC&FFP	74.61	74.56	-	-
Ours w/o FFP	76.47	76.41	**85.94**	55.36
Ours w/o CIC	76.37	76.36	-	-

In the AUI subtask, we used all framework modules mentioned in sections 3.2-3.4 simultaneously. However, in the ARI subtask, Causal Inference from ChatGPT is not used. That is because, even with strong constraints added, when inferring the argument relationship, a considerable part of the reasoning strings generated by the ChatGPT and T5 models still exceed the mapping vocabulary. The experimental results are shown in the Table 4.11. Our proposed framework exceeds the comparative baselines on both tasks. Since the GPT-4 model cannot be fine-tuned, it does not perform well on specific fine-grained financial analysis tasks, even worse than BERT-like models. Our proposed framework also achieves the first F1-scores in ARI subtask and the third F1-scores in AUI subtask of NTCIR-17 FinArg-1, respectively. Tasks details can refer to [107].

4.4.2 Ablation Experiments for Financial Argument Analysis

This section provides the ablation study for the proposed framework, as shown in the lower part of Table 4.11. In the AUI task, if only Prefix Instruction Fine-tuning

(PIF) is used instead of Financial Further Pre-training (FFP) and Causal Inference from ChatGPT (CIC), the obtained Macro-F1 score is 74.56, which is only slightly higher than the original T5 model, not as good as other SOTA LLMs model. If FFP or CIC is not used, the effect of the model will slightly decrease.

In the ARI task, we observed analogous results. Notably, in the absence of FFP, the model's Micro-F1 score for the ARI task exhibits an increase. However, this is accompanied by a significant decline in the Macro-F1 score. This underscores that the FFP module notably enhances the model's stability, especially when handling imbalanced datasets. For all ablation studies, we employed a substantial max length to mitigate the potential influence of the mapping vocabulary issue. A more in-depth discussion on max length can be found in the following sections.

4.4.3 Long or Short Instructions

When not using CIC, for each subtask, we use one long instruction and one short instruction for each subtask respectively, as mentioned in Table 3.1. At this point, we observed that whether to use FFP will have a significant impact on the inference results, as shown in Fig. 4.5. In general, shorter prompts give better results if FFP is used, on the other hand, longer prompts are better if FFP is not used. This could be attributed to the model acquiring relevant background knowledge during further pre-training. Without this knowledge, a more specific instruction-guided approach might be necessary during fine-tuning to produce the appropriate response.

4.4.4 Category Judgment with LLMs Supervision

In the actual inference process, when the output of the T5 model exceeds the max length parameter, it will be truncated. This situation can usually be avoided by setting max length greater than 256. But reducing the max length of the T5 model can exponentially speed up the inference, therefore, we always prefer a smaller max length. In order to solve this problem, we propose the Category Judgment method, which is to re-enter the

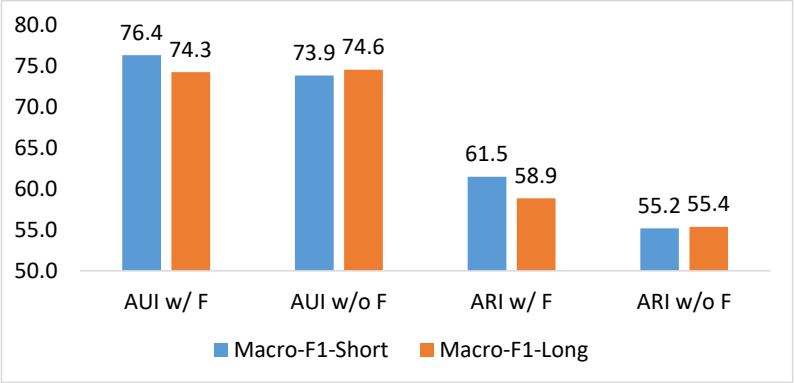

Fig. 4.5. Long/short instructions act on inference results.

outputs outside the mapping vocabulary into the fine-tuned model for further reasoning.

Some examples of case study and category judgment are listed in Table 4.12. Overall, the shorter the input sentence and the fewer compound arguments it contains, the more accurate the output of the model will be. The wrong predictions are concentrated in the case when the input sentence contains some complex ideas. When max length was set to be 40, with the Category Judgment method, most of the output exceeding the max length can be accurately identified. We also observed that some incorrectly predicted sentences leads to correct results after Re-Output, as shown in the third example in Talbel 4.12. Based on these results, we fond that after category judgment, the prediction accuracy of the model can reach a comparable level to that of using a large max length, thus greatly saving the computing resources of inference.

4.5 Conclusion

In conclusion, section 4 provides a detailed account of the experimental evaluations conducted using our proposed framework, encompassing five subtasks across four diverse datasets. These tasks, namely DQ, ND, MEP, ARI and AUI, serve as comprehensive test beds to assess the framework's efficacy in varied contexts.

The experimental results demonstrate the framework's robustness and adaptability. For instance, in DQ and ND subtasks, the framework showed a remarkable ability to dis-

Tab. 4.12. Case study and category judgment examples. Words in curly brackets indicate truncated parts when max length was set to be 40. Re-Output is the result of re-entering the output into the model with a larger max length. Green ticks represent correct predictions and red crosses represent incorrect predictions.

Predictions	Sentences
Output: Premise✓ **Re-Output:** Premise✓	**Input:** "And in that context, of course, they're lifting and shifting some of the older workloads, but they're modernizing the entire business process flow." **Output:** "This sentence provides evidence to support the argument by providing an example of how the argument is being implemented. It explains the action being taken to achieve the goal, which is to lift and shift older workloads, {and to modernize the entire business process flow. In summary, the sentence is a premise.}" **Re-Output:** "This sentence provides evidence to support the argument by providing an example of how the argument is being implemented. It does not make a statement that is meant to be accepted as true, but rather provides evidence to support the argument. In summary, the sentence is a premise."
Output: Claim✓ **Re-Output:** Claim✓	**Input:** "And in this particular case, now back to commenting on this one, we're hopeful that over time the restrictions that we're seeing are loosened because innovation really requires freedom to collaborate and communicate, and I know that that is a major focus there." **Output:** "This sentence is making a statement about the potential outcome of the restrictions that are being seen. It is expressing a hope that the restrictions will be loosened, and that this is a major focus of the argument{. In summary, the sentence is a claim.}" **Re-Output:** "This sentence is making a statement about the potential outcome of the restrictions being seen, and is expressing a hope that the restrictions will be loosened. This is a statement of opinion, and is therefore a claim. In summary, the sentence is a claim."

cern nuanced aspects of conversations and text segments. In the realms of MEP subtask, the framework adeptly handled the complexities of human emotions in text, showcasing its fine-grained analytical capabilities. Similarly, in Financial Argument Analysis, the framework's effectiveness in processing and interpreting specialized, domain-specific content was evident.

The section also delves into critical discussions that shed light on the underlying

Tab. 4.13. Case study and category judgment examples (continuation).

Predictions	Sentences
Output: Claim× **Re-Output:** Premise✓	**Input:** "I also really like what I've seen with the iPhone SE and the fact that it's opening the door to customers that we weren't reaching before and likely convincing some people to upgrade that wanted a smaller form factor but wanted to stay with iPhone, and so they were waiting for the iPhone SE." **Output:** "This sentence is making a claim about the success of the iPhone SE and the potential for it to convince customers to upgrade to a smaller form factor. It is not providing any evidence or facts to support this {claim, but is instead making a statement about the potential of the iPhone SE. In summary, the sentence is a claim.}" **Re-Output:** "This sentence provides evidence to support the argument by providing an example of a success of the iPhone SE and the potential for it to convince customers to upgrade to a smaller form factor. It does not make a statement about the argument itself, but rather provides evidence to support the argument. In summary, the sentence is a premise."
Claim✓	**Input:** "And one important way we drive selection is through FBA, and so we continue to be very pleased with the progress we're making in FBA." **Output:** "This sentence is making a statement about the progress that is being made inFBA, and is therefore a claim. In summary, the sentence is a claim."

mechanics and strategic choices in the framework's design. The selection of the pre-trained Transformer model was pivotal, highlighting the trade-offs between various models and their impact on the framework's performance. The debate between Neuro and Symbolic approaches underscored the importance of integrating logical reasoning with neural network-based learning. Insights into what the LDA learned revealed the depth of thematic understanding the framework could achieve.

Furthermore, the examination of long versus short instructions provided valuable findings on how instructional length influences model performance. Lastly, the discussion on category judgment with LLMs supervision brought to the forefront the role of large language models in enhancing the framework's accuracy and interpretability.

Overall, the experimental section not only validates the effectiveness of the pro-

posed framework but also opens avenues for future research. It highlights the potential of combining advanced NLP techniques with logical reasoning and large language models, setting a precedent for more sophisticated and interpretable sentiment analysis tools in various domains.

5 Conclusion and Future Work

5.1 Conclusion

In this paper, we have undertaken an in-depth exploration of text sentiment analysis, particularly focusing on the challenges posed by the predominant use of Deep Learning (DL) methodologies and Large Language Models (LLMs) in this domain. Despite the remarkable advancements achieved through these models, they often fall short in terms of interpretability and require substantial computational resources and extensively annotated datasets. These limitations not only impose financial and time constraints but also hinder a deeper understanding of the decision-making processes of these models.

Our research presents a groundbreaking, interpretative framework for text sentiment analysis. This framework is distinctive for its cost-efficiency and high precision, addressing the pivotal needs of the current sentiment analysis landscape. The core of our framework is built upon three innovative components. (1) Enriched Semantic Layer: We merge the unsupervised topic clustering LDA matrix with the hidden expression matrix generated by Transformer-like models. This integration significantly enhances the semantics of the hidden layer, enabling the model to discern deeper and more nuanced sentiments within the text. This step is crucial in bridging the gap between the raw data and its interpretability, allowing for a more profound understanding of the underlying emotional content; (2) Symbolic Logic Integration: Our framework incorporates symbolic logic systems, such as Real Logic and LTN, translating the often opaque operations of deep learning models into a format that is more logical and understandable. This integration is a leap towards demystifying the complex internal workings of these models, thereby rendering them significantly more interpretable. (3) Large Language Model Supervision: We utilize sophisticated language models, like ChatGPT, in a teacher-student dynamic. The 'teacher model' generates target text, serving as a benchmark to evaluate and guide the 'student model'. This approach not only allows our framework to glean insights from state-of-the-art models but also does so without inheriting their inherent opacity. We

further enhance our framework's performance and efficiency through techniques such as Prefix Instruction Fine-tuning, within-Domain Further Fine-tuning, and Task-specific Decoding. By incorporating these strategies, our methodology prioritizes simplicity and transparency while leveraging domain-specific knowledge. Initial results from our framework indicate a harmonious blend of interpretability and high performance, positioning it as a compelling alternative to existing deep learning-centric models.

To substantiate the efficacy of our proposed framework, we conducted extensive experiments on multiple sentiment analysis subtasks, including Weibo emotion detection, financial argument analysis, and human-machine customer service dialogue satisfaction evaluation. The framework's performance was outstanding, outperforming state-of-the-art baselines in various subtasks. Notably, it achieved first place in both the NTCIR-16 DialEval-2 and NTCIR-17 FinArg-1 tasks, underscoring its superiority in practical applications.

In conclusion, this research marks a significant stride towards developing more transparent, efficient, and accessible tools for sentiment analysis. The hybrid approach we have adopted, which skillfully combines interpretability with high performance, sets a new paradigm in the field. Our work not only contributes a novel perspective to sentiment analysis but also lays the groundwork for future innovations in creating more reliable, comprehensible, and user-friendly sentiment analysis methodologies. As we continue to refine and expand upon our framework, we are optimistic about its potential to revolutionize the field, making advanced sentiment analysis more attainable and insightful for a broader range of users.

5.2 Future Work

Building on the accomplishments and insights given from the research of this paper, several avenues for future work can be considered to further enhance and extend the capabilities of our Interpretable Text Sentiment Analysis Framework. These potential directions include:

Cross-Domain Adaptability: Future work could focus on adapting and testing the framework across a wider range of domains and text genres. This would involve tailoring the framework to handle the unique linguistic and sentimental nuances of texts from various fields such as healthcare, legal, educational, and technical domains.

Multilingual and Cross-Cultural Analysis: Expanding the framework to support multiple languages and cultural contexts is crucial. This would involve training and fine-tuning the model on diverse linguistic datasets and incorporating cultural context understanding to improve sentiment analysis accuracy in a global context.

Integration with Emerging Technologies: Exploring the integration of the framework with cutting-edge technologies such as Voice Recognition [108] and Computer Vision [109] could open up new applications. For instance, extending the multimodal capabilities [110] of the framework would significantly broaden the framework's applicability.

Advancements in Symbolic Logic Integration: Further research could delve into more advanced symbolic logic systems and their integration with neural networks. This could include exploring newer logic paradigms (such as PyReson) or enhancing the existing LTN opproach for better interpretability and accuracy.

Acknowledgement

I completed my PhD research after three years' study. First of all, I would like to express my sincere thanks to my supervisor Prof. Ren Fuji, for teaching me carefully. While giving me professional and comprehensive academic guidance, Prof. Ren also created a good learning environment and a strong learning atmosphere for me. His knowledgeable, approachable, rigorous academic image and practical work style left a deep impression on me, and he is a role model for my future career and life.

Secondly, I would like to thank all the teachers and students who helped me in my research life. I would like to express my gratitude to Prof. Shishibori, Prof. Fuketa, Prof. Nagata and Prof. Matsumoto, who carefully reviewed this thesis. They gave me great help in reviewing, guiding, and revising my thesis. I am also very grateful to Prof. Kang Xin. In the past three years, he has actively organized and discussed various academic issues, which broadened my research horizon, inspired me, and gave me more research perspectives and research ideas. Besides, the students in the research group not only answered my doubts in study, but also gave me a lot of care and help in life.

Finally, I would like to thank my family members, especially my wife Wang Liqing. They gave me spiritual and material support during my PhD research, which is a strong logistical support for me. And I would like to thank all the leaders, teachers, counselors and students of Tokushima University for their care and support.

References

[1] Marouane Birjali, Mohammed Kasri, and Abderrahim Beni-Hssane. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, 226:107134, 2021.

[2] Christina B Azodi, Jiliang Tang, and Shin-Han Shiu. Opening the black box: interpretable machine learning for geneticists. *Trends in genetics*, 36(6):442–455, 2020.

[3] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large language models, 2023.

[4] Rosalind W Picard. Affective computing. 1997. *Google Scholar Google Scholar Digital Library Digital Library*, 1997.

[5] Hanchen Jiang, Peng Lin, and Maoshan Qiang. Public-opinion sentiment analysis for large hydro projects. *Journal of Construction Engineering and Management*, 142(2):05015013, 2016.

[6] Yongfeng Zhang. Incorporating phrase-level sentiment analysis on textual reviews for personalized recommendation. In *Proceedings of the eighth ACM international conference on web search and data mining*, pages 435–440, 2015.

[7] Stefan Scherer, Michael Glodek, Georg Layher, Martin Schels, Miriam Schmidt, Tobias Brosch, Stephan Tschechne, Friedhelm Schwenker, Heiko Neumann, and Günther Palm. A generic framework for the inference of user states in human computer interaction. *Journal on Multimodal User Interfaces*, 6(3-4):117–141, 2012.

[8] Hong Mo, Jie Wang, Xuan Li, and Zhanlin Wu. Linguistic dynamic modeling and analysis of psychological health state using interval type-2 fuzzy sets. *IEEE/CAA Journal of Automatica Sinica*, 2(4):366–373, 2015.

[9] Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. Style transfer in text: Exploration and evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[10] Ke Wang, Hang Hua, and Xiaojun Wan. Controllable unsupervised text attribute transfer via editing entangled latent representation. In *Advances in Neural Information Processing Systems*, pages 11034–11044, 2019.

[11] Ke Wang and Xiaojun Wan. Sentigan: Generating sentimental texts via mixture adversarial networks. In *IJCAI*, pages 4446–4452, 2018.

[12] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[13] Yunqing Xia, Erik Cambria, Amir Hussain, and Huan Zhao. Word polarity disambiguation using bayesian model and opinion-level features. *Cognitive Computation*, 7(3):369–380, 2015.

[14] Hao Fei, Yue Zhang, Yafeng Ren, and Donghong Ji. Latent emotion memory for multi-label emotion classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7692–7699, 2020.

[15] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[16] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[17] Nourah Alswaidan and Mohamed El Bachir Menai. A survey of state-of-the-art approaches for emotion recognition in text. *Knowledge and Information Systems*, 62(8):2937–2987, 2020.

[18] Jiawen Deng and Fuji Ren. A survey of textual emotion recognition and its challenges. *IEEE Transactions on Affective Computing*, 2021.

[19] Erik Cambria, Yang Li, Frank Z Xing, Soujanya Poria, and Kenneth Kwok. Senticnet 6: Ensemble application of symbolic and subsymbolic ai for sentiment analysis. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 105–114, 2020.

[20] Md Shad Akhtar, Asif Ekbal, and Erik Cambria. How intense are you? predicting intensities of emotions and sentiments using stacked ensemble [application notes]. *IEEE Computational Intelligence Magazine*, 15(1):64–75, 2020.

[21] Mohammad Ehsan Basiri, Shahla Nemati, Moloud Abdar, Erik Cambria, and U Rajendra Acharya. Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis. *Future Generation Computer Systems*, 115:279–294, 2021.

[22] Hu Huang, Bowen Zhang, Liwen Jing, Xianghua Fu, Xiaojun Chen, and Jianyang Shi. Logic tensor network with massive learned knowledge for aspect-based sentiment analysis. *Knowledge-Based Systems*, 257:109943, 2022.

[23] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1-2):177–196, 2001.

[24] Luciano Serafini and Artur d'Avila Garcez. Logic tensor networks: Deep learning and logical reasoning from data and knowledge. *arXiv preprint arXiv:1606.04422*, 2016.

[25] Samy Badreddine, Artur d'Avila Garcez, Luciano Serafini, and Michael Spranger. Logic tensor networks. *Artificial Intelligence*, 303:103649, 2022.

[26] Tommaso Carraro. LTNtorch: PyTorch implementation of Logic Tensor Networks, mar 2022.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[28] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[29] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[30] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[31] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.

[32] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[33] Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, et al. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, page 100017, 2023.

[34] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[35] OpenAI. Gpt-4 technical report, 2023.

[36] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[37] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[38] Rosalind W Picard. *Affective computing*. MIT press, 2000.

[39] Mingxuan Hu and Min He. Non-parallel text style transfer with domain adaptation and an attention model. *Applied Intelligence*, 51(7):4609–4622, 2021.

[40] Rui Zhang, Zhenyu Wang, Kai Yin, and Zhenhua Huang. Emotional text generation based on cross-domain sentiment transfer. *IEEE Access*, 7:100081–100089, 2019.

[41] Zheng Li, Ying Wei, Yu Zhang, and Qiang Yang. Hierarchical attention transfer network for cross-domain sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[42] Kai Zhang, Hefu Zhang, Qi Liu, Hongke Zhao, Hengshu Zhu, and Enhong Chen. Interactive attention transfer network for cross-domain sentiment classification. In

Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 5773–5780, 2019.

[43] Peter D Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. *arXiv preprint cs/0212032*, 2002.

[44] Xin Kang, Fuji Ren, and Yunong Wu. Exploring latent semantic information for textual emotion recognition in blog articles. *IEEE/CAA Journal of Automatica Sinica*, 5(1):204–216, 2017.

[45] Fuji Ren and Lei Wang. Sentiment analysis of text based on three-way decisions. *Journal of Intelligent & Fuzzy Systems*, 33(1):245–254, 2017.

[46] Wouter Van Atteveldt, Mariken ACG van der Velden, and Mark Boukes. The validity of sentiment analysis: Comparing manual annotation, crowd-coding, dictionary approaches, and machine learning algorithms. *Communication Methods and Measures*, 15(2):121–140, 2021.

[47] Fuji Ren and Kazuyuki Matsumoto. Semi-automatic creation of youth slang corpus and its application to affective computing. *IEEE Transactions on Affective Computing*, 7(2):176–189, 2015.

[48] Milan Tripathi. Sentiment analysis of nepali covid19 tweets using nb svm and lstm. *Journal of Artificial Intelligence*, 3(03):151–168, 2021.

[49] Baris Ozyurt and M Ali Akcayol. A new topic modeling based approach for aspect extraction in aspect based sentiment analysis: Ss-lda. *Expert Systems with Applications*, 168:114231, 2021.

[50] Fuji Ren and Jiawen Deng. Background knowledge based multi-stream neural network for text classification. *Applied Sciences*, 8(12):2472, 2018.

[51] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer, 2019.

[52] Fei Ding, Xin Kang, Shun Nishide, Zhijin Guan, and Fuji Ren. A fusion model for multi-label emotion classification based on bert and topic clustering. In *International Symposium on Artificial Intelligence and Robotics 2020*, volume 11574, pages 98–111. SPIE, 2020.

[53] Chenyang Huang, Amine Trabelsi, and Osmar R Zaïane. Ana at semeval-2019 task 3: Contextual emotion detection in conversations through hierarchical lstms and bert. *arXiv preprint arXiv:1904.00132*, 2019.

[54] Kai Zhang, Qi Liu, Hao Qian, Biao Xiang, Qing Cui, Jun Zhou, and Enhong Chen. Eatn: An efficient adaptive transfer network for aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):377–389, 2021.

[55] Jianfei Yu, Luis Marujo, Jing Jiang, Pradeep Karuturi, and William Brendel. Improving multi-label emotion classification via sentiment classification with dual attention transfer network. ACL, 2018.

[56] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Advances in neural information processing systems*, 31, 2018.

[57] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*, 2019.

[58] Kyle Hamilton, Aparna Nayak, Bojan Božić, and Luca Longo. Is neuro-symbolic ai meeting its promise in natural language processing? a structured review. *arXiv preprint arXiv:2202.12205*, 2022.

[59] Fa-liang HUANG, Chao-xiong LI, Chang-an YUAN, Yan WANG, and Zhi-qiang YAO. Mining sentiment for web short texts based on tscm model. *ACTA ELECTONICA SINICA*, 44(8):1887, 2016.

[60] Wei Li, Luyao Zhu, Rui Mao, and Erik Cambria. Skier: A symbolic knowledge integrated model for conversational emotion recognition. 2023.

[61] Tailin Wu, Megan Tjandrasuwita, Zhengxuan Wu, Xuelin Yang, Kevin Liu, Rok Sosic, and Jure Leskovec. Zeroc: A neuro-symbolic model for zero-shot concept recognition and acquisition at inference time. *Advances in Neural Information Processing Systems*, 35:9828–9840, 2022.

[62] Erik Cambria, Qian Liu, Sergio Decherchi, Frank Xing, and Kenneth Kwok. Senticnet 7: A commonsense-based neurosymbolic ai framework for explainable sentiment analysis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3829–3839, 2022.

[63] Dyuman Aditya, Kaustuv Mukherji, Srikar Balasubramanian, Abhiraj Chaudhary, and Paulo Shakarian. Pyreason: Software for open world temporal logic. *arXiv preprint arXiv:2302.13482*, 2023.

[64] Thilo Rieg, Janek Frick, Hermann Baumgartl, and Ricardo Buettner. Demonstration of the potential of white-box machine learning approaches to gain insights from cardiovascular disease electrocardiograms. *PloS one*, 15(12):e0243615, 2020.

[65] Hui Wen Loh, Chui Ping Ooi, Silvia Seoni, Prabal Datta Barua, Filippo Molinari, and U Rajendra Acharya. Application of explainable artificial intelligence for healthcare: A systematic review of the last decade (2011–2022). *Computer Methods and Programs in Biomedicine*, page 107161, 2022.

[66] Marina Martins. Analysis of high school students' argumentative dialogues in different modelling situations. *Science & Education*, pages 1–38, 2022.

[67] Hajra Waheed, Saeed-Ul Hassan, Naif Radi Aljohani, Julie Hardman, Salem Alelyani, and Raheel Nawaz. Predicting academic performance of students from vle big data using deep learning models. *Computers in Human behavior*, 104:106189, 2020.

[68] Isaac Machorro-Cano, Giner Alor-Hernández, Mario Andrés Paredes-Valverde, Lisbeth Rodríguez-Mazahua, José Luis Sánchez-Cervantes, and José Oscar Olmedo-Aguirre. Hems-iot: A big data and machine learning-based smart home system for energy saving. *Energies*, 13(5):1097, 2020.

[69] Ivan Cvitić, Dragan Peraković, Marko Periša, and Brij Gupta. Ensemble machine learning approach for classification of iot devices in smart home. *International Journal of Machine Learning and Cybernetics*, 12(11):3179–3202, 2021.

[70] Niklas Bussmann, Paolo Giudici, Dimitri Marinelli, and Jochen Papenbrock. Explainable ai in fintech risk management. *Frontiers in Artificial Intelligence*, 3:26, 2020.

[71] Xinsong Ma, Feifei Zheng, and Denglin Tang. Identifying the head-and-shoulders pattern using financial key points and its application in consumer electronic stocks. *IEEE Transactions on Consumer Electronics*, 2023.

[72] Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. Aspect sentiment quad prediction as paraphrase generation. *arXiv preprint arXiv:2110.00796*, 2021.

[73] Tianhao Gao, Jun Fang, Hanyu Liu, Zhiyuan Liu, Chao Liu, Pengzhang Liu, Yongjun Bao, and Weipeng Yan. Lego-absa: A prompt-based task assemblable unified generative framework for multi-task aspect-based sentiment analysis. In *Proceedings of the 29th international conference on computational linguistics*, pages 7002–7012, 2022.

[74] Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. A survey on aspect-based sentiment analysis: tasks, methods, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[75] Jordan J Bird, Anikó Ekárt, and Diego R Faria. Chatbot interaction with artificial intelligence: human data augmentation with t5 and language transformer ensemble for text classification. *Journal of Ambient Intelligence and Humanized Computing*, 14(4):3129–3144, 2023.

[76] Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. Exploring listwise evidence reasoning with t5 for fact verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 402–410, 2021.

[77] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance, 2023.

[78] Ling Luo, Xiang Ao, Feiyang Pan, Jin Wang, Tong Zhao, Ningzi Yu, and Qing He. Beyond polarity: Interpretable financial sentiment analysis with hierarchical query-driven attention. In *IJCAI*, pages 4244–4250, 2018.

[79] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*, 2019.

[80] Keane Ong, Wihan van der Heever, Ranjan Satapathy, Gianmarco Mengaldo, and Erik Cambria. Finxabsa: Explainable finance through aspect-based sentiment analysis. *arXiv preprint arXiv:2303.02563*, 2023.

[81] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`, 2022.

[82] Changqin Quan and Fuji Ren. A blog emotion corpus for emotional expression analysis in chinese. *Computer Speech & Language*, 24(4):726–749, 2010.

[83] Zhaohao Zeng and Tetsuya Sakai. Dch-2: A parallel customer-helpdesk dialogue corpus with distributions of annotators' labels. *arXiv preprint arXiv:2104.08755*, 2021.

[84] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[85] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[86] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

[87] Qi Zhu, Yuxian Gu, Lingxiao Luo, Bing Li, Cheng Li, Wei Peng, Minlie Huang, and Xiaoyan Zhu. When does further pre-training mlm help? an empirical study on task-oriented dialog pre-training. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 54–61, 2021.

[88] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[89] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[90] Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050*, 2023.

[91] Tetsuya Sakai. Evaluating evaluation measures for ordinal classification and ordinal quantification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2759–2769, 2021.

[92] Tetsuya Sakai. Comparing two binned probability distributions for information access evaluation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1073–1076, 2018.

[93] Bing Liu. *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge university press, 2020.

[94] Deyu Zhou, Yang Yang, and Yulan He. Relevant emotion ranking from text constraint with emotion relationships. Association for Computational Linguistics, 2018.

[95] Xin Kang, Yunong Wu, and Fuji Ren. Tua1 at the ntcir-15 dialeval task. In *Proceedings of the 15th NTCIR Conference on Evaluation of Information Access Technologies*, pages 53–56, 2020.

[96] Sijie Tao and Tetsuya Sakai. Overview of the ntcir-16 dialogue evaluation (dialeval-2) task. *Proceedings of NTCIR-16. to appear*, 2022.

[97] Xiang Zhang and Yann LeCun. Which encoding is the best for text classification in chinese, english, japanese and korean? *arXiv preprint arXiv:1708.02657*, 2017.

[98] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vish-

wanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[99] Fuji Ren and Ning Liu. Emotion computing using word mover' s distance features based on ren_cecps. *PloS one*, 13(4):e0194136, 2018.

[100] Jiawen Deng and Fuji Ren. Hierarchical network with label embedding for contextual emotion recognition. *Research*, 2021.

[101] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2, 2019.

[102] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, 2022.

[103] Heng-Yu Lin, Eugene Sy, Tzu-Cheng Peng, Shih-Hsuan Huang, and Yung-Chun Chang. Tmunlp at the ntcir-17 finarg-1 task. In *Proceedings of the 17th NTCIR conference on evaluation of information access technologies. https://doi. org/10.20736/0002001286*, 2023.

[104] Shaopeng Tang and Lin Li. Idea at the ntcir-17 finarg-1 task: Argument-based sentiment analysis. In *Proceedings of the 17th NTCIR conference on evaluation of information access technologies. https://doi. org/10.20736/0002001276*, 2023.

[105] Swagata Chakraborty, Anubhav Sarkar, Dhairya Suman, Sohom Ghosh, and Sudip Kumar Naskar. Lipi at the ntcir-17 finarg-1 task: Using pre-trained language models for comprehending financial arguments. In *Proceedings of the 17th NTCIR conference on evaluation of information access technologies. https://doi. org/10.20736/0002001281*, 2023.

[106] Lefteris Loukas, Manos Fergadiotis, Ilias Chalkidis, Eirini Spyropoulou, Prodromos Malakasiotis, Ion Androutsopoulos, and Georgios Paliouras. Finer: Financial numeric entity recognition for xbrl tagging. *arXiv preprint arXiv:2203.06482*, 2022.

[107] Chung-Chi Chen, Chin-Yi Lin, Chr-Jr Chiu, Hen-Hsen Huang, Alaa Alhamzeh, Yu-Lieh Huang, Hiroya Takamura, and Hsin-Hsi Chen. Overview of the ntcir-17 finarg-1 task: Fine-grained argument understanding in financial analysis. In *Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies, Tokyo, Japan*, 2023.

[108] Attila Andics, James M McQueen, Karl Magnus Petersson, Viktor Gál, Gábor Rudas, and Zoltán Vidnyánszky. Neural mechanisms for voice recognition. *Neuroimage*, 52(4):1528–1540, 2010.

[109] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, Eftychios Protopapadakis, et al. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[110] Ankita Gandhi, Kinjal Adhvaryu, Soujanya Poria, Erik Cambria, and Amir Hussain. Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions. *Information Fusion*, 91:424–444, 2023.