Advancements and Applications in Natural Language Processing: A Comprehensive Literature Review

Xianqing Zeng, Haoxuan Qin, Wenyan Chen, Ruiyao Chen, Peizhi Tang, Qingchang Hu

***Abstract*—This review explores the evolution and current trends in Natural Language Processing (NLP). It discusses the transition from early machine translation to advanced models. Key areas like coreference resolution, discourse parsing, and summarization generation are examined, alongside the integration of NLP in dialogue systems, with a focus on bias handling and factuality assurance. The report also delves into the challenges and future directions of NLP, highlighting its impact on understanding natural languages and human cognition, and its growing significance across various disciplines.**

***Index Terms*—Natural Language Processing, Conference resolution, Discourse parsing, Discourse structure, Summary generation, Dialogue systems, Chatbots, Deep learning**

I. Introduction

NATURAL language processing (NLP) is a branch of artificial intelligence that aims to enable machines to understand and generate natural language. NLP has a wide range of applications, such as machine translation, sentiment analysis, text summarization, question answering, and dialogue systems. The development of NLP has been greatly accelerated by the availability of large-scale datasets and powerful neural network models, such as BERT, GPT-4, or MEGATRON. These models can learn from massive amounts of text data and perform various NLP tasks, we mentioned above. However, these models also pose new challenges and questions for both NLP and linguistics. For example, how well do these models capture the linguistic knowledge and rules of natural languages? How do these models affect the evolution and diversity of natural languages? How do these models influence the way people think and communicate with language? How do these models relate to the philosophical issues of language, such as meaning, truth, or intentionality?

These are some of the topics that NLP and linguistics researchers are exploring and debating in the current era of large-scale language models. The connection between NLP and linguistics is not only technical but also theoretical and ethical. The development of NLP has the potential to enrich our understanding of natural languages and human cognition, but also to raise new problems and dilemmas that require careful consideration and collaboration among different disciplines.

A. History

The history of NLP began as translation machine after the World War II, when people at that time hoped to create a machine that can can be used to translate between different languages automatically given the rising need of communication especially for diplomacy and military. However, the work was not that easy as what the people expected. Noam Chomsky, a linguistic expert explained that the model at that time did not distinguish sentences that are grammatically correct but semantically unreasonable and sentences that are both incorrect. For example, consider 2 ”artificial” sentences, ”Colorless green ideas sleep furiously”, and ”Furiously sleep ideas green colorless”. From a human perspective, the former one is grammatically correct but nonsense, and the latter one is just nonsense. Chomsky was unsatisfied that the machine can not obtain the same result. Also, given that the limitation of computer at that time (low memory and storage, slow process rate), a single sentence would take 7 minutes to analyse a long sentence even for the best machine with the best algorithm. [1]

At the time around 1960s, two different approaches to NLP emerged: symbolic and stochastic. Symbolic NLP, also known as rule-based NLP, focused on formal languages and syntax generation. It was mainly pursued by linguists and computer scientists who saw it as a way to advance artificial intelligence. Rule-based approach involves applying a particular set of rules or patterns to capture specific structures, extract information, or perform tasks such as text classification and so on. Some common rule-based techniques include regular expressions and pattern matches. Stochastic NLP, on the other hand, used statistical and probabilistic methods to deal with optical character recognition and text pattern recognition problems. Its aim is to resolve difficulties that arise because longer sentences are highly ambiguous when processed with realistic grammars, yielding thousands or millions of possible analyses.

By the early 1980s, it became apparent that building well-founded, predictable, and extensible NLP systems was more difficult than initially anticipated. This led to a shift in focus towards more structured and principled approaches. This period saw the development of powerful general-purpose processors, such as SRI’s Core Language Engine, which could support application systems with operational power and potential for superior performance. The emphasis was on a declarative approach, unification, and syntax-driven compositional interpretation into logical forms. The grammatico-logical approach which is featured by, a focus on grammar and logic, led to the adoption of predicate calculus-style meaning representations, even in cases where the processes delivering these representations were more informal. [2] [3]

Afterwards, there has been a significant move towards using statistical methods in NLP, driven by the success of these approaches in speech processing and the increasing availability of large text corpora. Also, researchers have focused on creating suitable formalisms for representing lexical information, tying it closely to syntactic and semantic processing and leveraging AI experience in knowledge representation. When both the technology of NLP and hardware were developing, modern computing technology has enabled the development of systems that combine language with other modes or media, such as graphics, although the impact of this combination on language processing remains an open question. [4]

It is seen that the state of NLP has seen improved understanding of computation implications, with progress in syntax and grammar characterization, and the development of various conceptual tools and techniques but even the most effective current systems before 2000s are still limited to narrow domains or tasks. The need for evaluation in user contexts is essential, especially for less demanding tasks like document retrieval. Despite the acceptance of highly modular architectures, challenges remain in determining the distribution of information and effort between linguistic and non-linguistic elements in a system.

After 2000, the field of NLP experienced significant growth and advancements, especially in the recent year. The advent of deep learning had a profound impact on NLP research and applications. The development of neural network architectures, such as recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, enabled the modeling of complex language structures and led to breakthroughs in tasks like language modeling, machine translation, and sentiment analysis. [4] Another evolution embedding model called Word2Vec came out in 2013. It uses neural networks to generate dense vector representations of words. These representations capture semantic and syntactic relationships, significantly improving the performance of various NLP tasks. [5] From 2015 to the present, Natural Language Processing (NLP) has experienced significant advancements, primarily driven by deep learning techniques, large-scale pre-trained models, and increased computational resources. Key breakthroughs include the introduction of transformer models like BERT, GPT, and T5, which have achieved remarkable performance on a wide range of NLP tasks. These models utilize self-attention mechanisms, unified text-to-text formats, and transfer learning, allowing them to be easily fine-tuned for specific tasks with limited training data. Additionally, efficient models like ALBERT and XLNet have been developed to reduce computational requirements while maintaining strong performance. The success of these pre-trained models has revolutionized NLP, enabling numerous practical applications and expanding the potential of language models. [6] [7] [8]

B. Two main categories

1) Natural Language Understanding (NLU):

Natural Language Understanding (NLU) refers to the capacity of machines to process, analyze, and produce human language effectively. This field encompasses various subdisciplines, including syntax, which focuses on the structure and grammar of language; semantics, which deals with the meaning of words, phrases, and sentences; and pragmatics, which examines language in context and its use in social interactions. The primary objective of NLU is to allow machines to understand and interpret human language in a way that is both meaningful and contextually relevant. This involves not just recognizing words and phrases, but also grasping the nuances, intentions, and emotions conveyed by the speaker or writer.

There are 3 key components for NLU:

1. **Syntax and Semantic Analysis**: One of the fundamental steps in NLU involves analyzing the syntax - the arrangement of words to form meaningful sentences, and semantics - the meaning that is conveyed. Syntax analysis involves parsing and tagging parts of speech, while semantic analysis involves understanding the implications and intended messages.
2. **Context Understanding and Sentiment Analysis**: Context plays a vital role in understanding language. NLU systems must decipher context to accurately interpret the meaning of words and phrases. Sentiment analysis, a part of NLU, involves identifying and categorizing opinions expressed in text, particularly to determine the writer's or speaker's attitude.
3. **Entity Recognition and Disambiguation**: This involves identifying and classifying key elements in text into predefined categories such as names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. Disambiguation is crucial to differentiate between words that have multiple meanings based on context.

2) Natural Language Generation (NLG):

Natural Language Generation (NLG) is the aspect of NLP that focuses on generating coherent and contextually relevant natural language text or speech from a structured data source. Unlike NLU, which interprets human language, NLG is about producing human-like language from machine-understood inputs. NLG plays a pivotal role in automating content creation, providing machines with the ability to articulate ideas, thoughts, or data in a way that is indistinguishable from human-generated content. This ability is crucial for applications ranging from report generation to conversational agents.

NLG is shaping the future of interactive storytelling and virtual assistants, providing a more natural and engaging user experience. It enables dynamic story generation and enhances virtual assistants to make them more conversational and responsive. The future of NLG is promising, with ongoing advancements in AI and machine learning. Emerging trends include more sophisticated context awareness, emotional intelligence in generated language, and seamless integration of NLG in various multimedia platforms.

II. Coreference Resolution

A. Introduction to Coreference Resolution

Coreference Resolution, a fundamental component of natural language processing (NLP), focuses on identifying and linking various expressions within a text that refer to the same real-world entity. This task is vital in enhancing the comprehension and interpretation of text, significantly contributing to the effectiveness of other NLP applications like machine translation, sentiment analysis, paraphrase detection, and summarization. Coreference Resolution encompasses two main sub-areas: Anaphora Resolution, which deals with connecting pronouns to their specific nouns or names, and the broader Coreference Resolution, which also includes identifying complex ”bridging relationships” between different expressions. The field has witnessed significant advancements, evolving from rule-based systems to sophisticated deep learning methodologies. This progress reflects the intricate nature of the task and its central role in achieving a comprehensive understanding of textual data in various NLP applications. [9]

B. Anaphora and Coreference: Definitions and Concepts

Coreference resolution, a critical component in the realm of Natural Language Processing (NLP), plays a pivotal role in enhancing the comprehension and processing of language by machines. It fundamentally deals with identifying words or phrases in a text that refer to the same entity in the world. This concept is integral to understanding the complexities and intricacies of human language, as it enables the construction of a coherent narrative in discourse by linking various parts of the text.

Anaphora, a specific subset of coreference, is particularly concerned with the relationship between pronouns and their antecedents – the nouns or names they refer to. This type of resolution helps in bridging the gap between different sentences or segments of a text, thereby allowing for a more holistic understanding. Anaphoric entities, together with co-referent entities, form a part of the broader domain of discourse parsing, which is essential for complete text comprehension.

The history of research in anaphora resolution within the NLP community reveals its complexity and the gradual progression of methodologies. Starting from rule-based systems in the early days to the more recent deep learning approaches, anaphora resolution has been a field of steady yet slow progress. This slow advancement underscores the intricate nature of the task, which requires a nuanced understanding of language and context.

The application of anaphora resolution spans various crucial fields within NLP, such as sentiment analysis, summarization, machine translation, and question answering. By resolving anaphoric references, these fields are endowed with the capability to extend their scope from an intra-sentential level to an inter-sentential level, thereby enhancing their overall effectiveness and applicability.

Understanding coreference and anaphora is vital for full text understanding in NLP, as it enables the construction of a more connected and coherent narrative in language processing. The evolution of methodologies in this field, from rule-based systems to advanced neural network applications, illustrates the ongoing research and development aimed at improving machine understanding of human language.

C. Challenges in Coreference Resolution

The challenges in Coreference Resolution (CR) are multifaceted and stem primarily from the complexity and variability of natural language, as well as the diverse operationalization of coreference in different datasets. Coreference resolution, an essential task in Natural Language Processing (NLP), involves identifying expressions within a discourse that refer to the same entity, concept, or event. It plays a pivotal role in constructing representations of natural language and is crucial for various NLP tasks like question answering, summarization, and machine translation. [10]

One of the primary challenges in CR arises from the varying ways in which coreference is realized in different datasets. These variations are due to factors such as annotation guidelines, task formats, and the choice of corpora. This diversity in operationalization affects the performance of CR models, as they often struggle to generalize across different types of coreference instances [10]. For instance, models trained on one dataset may perform poorly on another due to differences in how coreference types, like generic mentions or copula predicates, are annotated [10].

CR models have evolved from initial rule-based approaches to current state-of-the-art models that utilize supervised training with pre-trained language model encoders. Despite advancements, these models still face challenges in handling various types of coreference, especially those requiring semantic knowledge [10]. Generalization remains a significant hurdle, as models often show decreased performance when evaluated on genres or types of coreference not included in their training datasets. This limitation highlights the need for models that can adapt across different operationalizations of coreference [10].

Further complicating matters, different datasets may have unique ways of annotating coreference types, such as exact matches, nested mentions, compound modifiers, generics, copula predicates, and pronominal anaphora. These variations necessitate a nuanced approach to training and evaluating CR models, as models trained on one set of annotation guidelines might not perform well on datasets with different guidelines [10].

Performance assessments of CR models also reveal that models trained on the same dataset tend to be more correlated in their correct predictions, suggesting a strong link between the training dataset's characteristics and the model's performance. This observation underscores the importance of considering dataset-specific factors when developing and evaluating CR models [10].

In summary, the challenges in Coreference Resolution are deeply intertwined with the variability of natural language and the diversity in dataset operationalization. Addressing these challenges requires a concerted effort to develop models that can adapt to different types of coreference and generalize effectively across diverse datasets.

D. Types of Anaphora in Coreference Resolution

The complexity of anaphora resolution (AR) in coreference resolution is highlighted by the diverse forms of references encountered in natural language. One significant challenge is the coverage issue, where most AR and coreference resolution (CR) algorithms target only specific reference types. Understanding these reference types is essential for grasping the scope of AR and CR.

Zero anaphora, common in prose and ornamental English, uses a gap in a phrase or clause to refer back to its antecedent, thereby involving multiple layers of interpretation. One anaphora, although less common, employs the word ”one” to refer to a previously mentioned antecedent. Demonstratives are used for comparisons with previously mentioned entities but are not explicitly specified in the text. Presuppositions involve pronouns like someone, anybody, and nobody, where the ambiguity in noun phrase reference adds complexity. Discontinuous sets or split anaphora occur when a pronoun refers to multiple antecedents, posing a challenge to algorithms that typically expect one-to-one mappings.

Contextual disambiguation in AR intersects with word sense disambiguation in CR, where the context determines the referent's real-world entity. Pronominal anaphora, a common type in everyday language and web data, includes indefinite, definite, and adjectival pronominal forms, each presenting unique identification challenges. Cataphora, the opposite of anaphora, points to entities that follow it in the text, and while less common, adds to the complexity of AR. Lastly, inferable or bridging anaphora involves references that imply but do not explicitly state their antecedents, requiring inferential reasoning based on context [9].

E. Linguistic Constraints for Anaphoric Resolution

In addressing the topic of linguistic constraints for anaphoric resolution in natural language processing (NLP), it is important to consider various recent studies and advances in this field. The task of resolving anaphora, particularly abstract anaphora, poses significant challenges for text understanding. However, with advancements in representation learning, these challenges are being increasingly addressed. A notable development in this area is the mention-ranking model that utilizes LSTM-Siamese Networks. This model is designed to learn how abstract anaphors relate to their antecedents, and it has been shown to outperform existing methods in shell noun resolution. Moreover, this model addresses the issue of limited training data by generating artificial anaphoric sentence–antecedent pairs. It has achieved benchmark results on the abstract anaphora subset of the ARRAU corpus, which is known for its complexity due to a mix of nominal and pronominal anaphors and a range of confounders. The model's strength lies in its ability to select syntactically plausible candidates and, when syntax is disregarded, to discriminate candidates based on deeper features [11].

This approach illustrates the evolving landscape of linguistic constraint application in anaphoric resolution, highlighting the importance of innovative machine learning techniques and the utilization of comprehensive, challenging corpora for model training and evaluation. As the field progresses, such developments are crucial in enhancing the accuracy and efficiency of NLP systems in interpreting and processing natural language.

F. Evaluation Metrics in Coreference Resolution

The evaluation of coreference resolution, particularly in cross-document (CD) scenarios, has historically been challenged by overly lenient practices, leading to inflated performance metrics. This issue has been addressed by proposing two key evaluation principles:

1. **Evaluation on Predicted Mentions:** Traditional methods often assume 'gold' entity and event mentions are given. However, real-world application demands models to perform on raw text, including automatic mention detection. This approach faces the challenge of handling singleton entities (mentioned only once) differently, necessitating the separation of mention detection from coreference linking evaluations [12].
2. **Addressing Lexical Ambiguity:** CD coreference resolution is complicated by lexical ambiguities, where the same event may be described differently across various documents, or different events may be referenced using similar terms. The standard ECB+ dataset, used for benchmarking, artificially simplifies this by clustering documents into subtopics and topics. This structure has been exploited by models to bypass the challenge of lexical ambiguity. To counter this, it's proposed that models should also report performance at the topic level, thus better reflecting real-world scenarios and conforming to the original intent of the ECB+ corpus [12].

   Empirical results show that adopting these more stringent evaluation criteria drastically affects performance. For instance, a competitive model's score dropped by 33 F1 points under these new evaluation principles compared to previous lenient practices, highlighting the need for future models to address these real-world complexities [12].

III. Discourse Parsing

A. Introduction to Discourse Parsing

Discourse Parsing, a crucial component within the field of Natural Language Processing (NLP), is dedicated to analyzing and deconstructing the structure of coherent sentences within a text. This process involves the identification of discourse units, such as sentences or clauses, within the text, and parsing their logical and rhetorical relationships. Through this mechanism, Discourse Parsing aids in uncovering the deeper structure and meaning of the text, providing essential insights for understanding and interpreting the content. This task is vitally important in enhancing the comprehension and interpretation of texts, significantly contributing to the effectiveness of other NLP applications. These include machine translation, sentiment analysis, paraphrase detection, and summarization, where accurate parsing of discourse structure plays a pivotal role in improving their performance and precision. The field has undergone significant development, evolving from rule-based systems to complex deep learning methodologies. This evolution reflects the intricate nature of the task and its central importance in achieving a comprehensive understanding of textual data across various NLP applications. [13] [14]

Discourse Parsing primarily encompasses three subfields: RST-style Discourse Parsing, PDTB-style Discourse Parsing, and Dialogue Discourse Parsing. RST-style parsing aims to parse documents into a hierarchical tree structure, addressing the elementary discourse units in the text and their interrelationships; PDTB-style parsing focuses on identifying and labeling the local discourse relations between two arguments, without forming a tree structure; while Dialogue Discourse Parsing deals with dialogues, building discourse dependency graphs to analyze the structure and relationships in multi-party conversations. Each style of Discourse Parsing has its unique methods and application scenarios, together constituting the complex and diverse research field of Discourse Parsing.

B. RST-style Discourse Parsing

Rhetorical Structure Theory (RST) is a framework used to represent the structure of a document. RST-style parsing aims to parse a document into a hierarchical tree structure, dealing with the elementary discourse units (EDUs) in the text and their interrelations. RST-style Discourse Parsing includes two main tasks: discourse segmentation and tree building. [15]

Discourse segmentation is the process of identifying the elementary discourse units (EDUs). In the RST tree, these are the leaf nodes. EDUs are interconnected through various discourse relations (such as cause-effect, contrast, elaboration), forming the structure of the text. The inner nodes are called a span that contains two or more adjacent EDUs. In RST, discourse relations are categorized into two types: hypotactic (mononuclear) and paratactic (multi-nuclear). In mononuclear relations, an inner node connects two EDU nodes, where the more salient node is termed as the nucleus and the other as the satellite. The nucleus unit is the primary part of the relation, while the satellite unit provides additional, supportive information. In paratactic relations, all nodes or spans are equally significant.

The tree-building task involves organizing these EDUs into an RST tree, entailing the structured representation of the text based on the segmented EDUs. This task includes the following subtasks: span prediction, nuclearity indication, and relation classification. Span prediction is a binary classification task aimed at predicting which EDUs or spans in the text should be merged together. This helps determine the structural hierarchy within the text. Nuclearity indication involves predicting which of the two EDUs or spans is the nucleus and which is the satellite. Relation classification aims to classify the specific rhetorical relations between the given two EDUs or spans. For example, these relations might be cause-effect, contrast, elaboration, etc. These tasks are crucial for understanding and analyzing the complex structure of texts, especially when dealing with complex documents, academic papers, or other professional texts. Through these analyses, a deeper understanding of the text's organizational structure and the logical relationships between its parts can be achieved. [16]

This framework emphasizes that the organization of text is not arbitrary but is purposefully structured to achieve specific communicative goals or intents. Different discourse relations and structures contribute to these communicative objectives. In understanding a text, RST considers discourse units and relationships at various levels. From individual sentences to entire text segments, different levels of structure collectively constitute the overall meaning of the text. RST-style parsing is crucial in understanding the organizational and argumentative structure of a text, which is vital for advanced NLP applications such as summarization, information extraction, and text generation. The Rhetorical Structure Theory provides a framework for understanding how different parts of a text contribute to the overall message and intent of the author, making it an important tool in the field of discourse analysis.

C. PDTB-style Discourse Parsing

Unlike methods based on Rhetorical Structure Theory (RST), PDTB in discourse parsing does not parse discourse into a hierarchical tree structure. Instead, it opts for a more linear and planar approach for annotating and analyzing discourse relations. This approach emphasizes intuitive and direct mapping of relations in actual texts, rather than establishing complex hierarchical structures. This method makes PDTB particularly suitable for texts without obvious hierarchical structures, such as news reports and informal dialogues.

In the PDTB processing workflow, the first step is text preprocessing, which includes tokenization, sentence splitting, and part-of-speech (POS) tagging. These steps lay the foundation for understanding the basic structure and components of the text, ensuring the accuracy and effectiveness of subsequent analysis.The next step is connective recognition, where PDTB distinguishes between explicit and implicit connectives. Explicit connectives, like 'because' or 'however,' can be directly identified from the text. In contrast, the recognition of implicit connectives requires unveiling the implicit, unexpressed discourse relations in the text, demanding a deeper understanding of the text's deeper meanings and context. This is followed by argument recognition, where analysts need to identify the arguments associated with each connective. These arguments are the core text fragments constituting the discourse relations. This involves determining the boundaries of the arguments and analyzing their internal structure and composition to ensure an accurate understanding of the discourse relations. Based on the recognition of connectives and related arguments, PDTB further classifies discourse relations into different types, such as causal or contrast relations. Additionally, PDTB provides extra information for discourse relations through attribution annotation, such as the source, certainty, and polarity of the relations, adding depth and richness to the analysis. Finally, PDTB employs a planar representation method, presenting discourse relations as linear connections between parts of the text, rather than building a hierarchical tree structure. This method is closer to the actual structure of the text, making the understanding and analysis of discourse relations more intuitive and operable.

The PDTB 1.0 discourse parser includes all modules of PDTB-style discourse parsing, such as connective classifier, argument labeler, explicit classifier, non-explicit classifier, and attribution span labeler. In the sense hierarchy of PDTB 2.0, discourse relations are divided into five categories: Explicit, Implicit, AltLex, Entel, and NoRel. In PDTB 3.0, AltLexC and Hypophora tokens are added. [17]

D. Dialogue Discourse Parsing

Dialogue Discourse Parsing is focused on in-depth processing and analysis of conversational content, aiming to understand and reveal the hidden structures and semantic relationships within dialogues. The initial step in this process involves breaking down complex dialogue data into smaller, more manageable units known as Elementary Discourse Units (EDUs). Each EDU, often encapsulating a complete idea or concept, could be an individual sentence or a segment of the dialogue.

The key step in Dialogue Discourse Parsing is predicting edges between these EDUs, which entails identifying and categorizing complex relationships between the units. In this context, EDUs are viewed as fundamental components of the dialogue. The edge detection process not only uncovers which EDUs are interconnected but also identifies the nature of these connections. Different edge types represent diverse relationship types, such as causal, contrastive, or supplementary relations.

Based on these insights, Dialogue Discourse Parsing aims to construct a model that accurately reflects the dialogue's flow and structure. This involves creating tree-like, graph-based, or other more complex structures, offering a robust viewpoint for a deeper understanding of the dialogue content. This technique is vital for enhancing dialogue systems, improving machine understanding of human language, and facilitating more natural and fluid human-machine interactions. Powered by deep learning and machine learning technologies, Dialogue Discourse Parsing demonstrates significant potential in capturing intricate dialogue structures and nuanced semantics, presenting new opportunities for advancement in the field of artificial intelligence. [18]

E. Applications and Future Trends of Discourse Parsing

Discourse Parsing has wide-ranging applications in various fields. These applications include, but are not limited to, information extraction, text summarization, sentiment analysis, question answering systems, machine translation, and natural language understanding. By accurately parsing the discourse structure of text, these applications can better comprehend and process complex language data. The field of discourse parsing is expected to continue evolving, especially driven by deep learning technologies, bringing new breakthroughs and enhancements to various natural language processing applications. Simultaneously, research in this field will face new challenges and opportunities.

Despite the progress made by deep learning methods in discourse parsing, challenges persist, such as effectively handling implicit discourse relationships, training efficient models on small datasets, and integrating various types of discourse relationships. These challenges also present new research opportunities for scholars. In the future, with further applications of deep learning, discourse analysis is expected to advance towards more precise and detailed directions. For example, the depth of understanding could be enhanced through multimodal analysis, combining text, speech, images, and other modalities. Additionally, the development of interactive discourse analysis will open up new possibilities for dialogue systems and intelligent assistants. Cross-linguistic and cross-cultural discourse analysis will better adapt to the global communication demands.

In conclusion, as a continually progressing technology, discourse analysis will play an increasingly vital role in understanding and processing complex textual information.

IV. Summary Generation

A. Extractive and Abstractive Summarization

1) Extractive Summarization:

This method extracts sentences, paragraphs, or phrases directly from the original text and combines them into a summary without rewriting or re-expressing the content. Extractive summarization process can be divided into three phases. [19]

The first one is pre-processing phase. There are many techniques, such as part of speech(POS) tagging, stop word filtering and stemming.

1. **Part of Speech(POS) Tagging**: It is the the process of specifying the words of text according to speech category such as noun, verbs, adverbs, adjectives etc.
2. **Filtering Stop Words**: Stop words are words which are filtered out before or after processing of text, it is fully non-objective depends upon the situation. For example, a, an, in, the can be considered as stop words and filtered from the text.
3. **Stemming**: Stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form. For example, removing from -ed, –ing, -s from verbs, using singular instead of plural noun, etc.

The second phase is to calculate the score of sentence, indicating the degree whether it belongs to summary or not, the following feautures are generally used in the score of sentences.

1. **Title Similarity**: A sentence has the highest score if it has the maximum number of words similar in the title. The title similarity is defined as

$$f1 = \frac{S \cap T}{t}$$

$$\begin{cases} S = \text{Set of words in sentnece} \\ T = \text{Set of words in title} \\ S \cap T = \text{Similar words in sentence and title} \end{cases}$$

2. **Sentence Position**: A sentence can be evaluated by its position in the text. For example, considering that there are more than three sentences in the paragraph.

$$f2 = \begin{cases} 3 \text{ for 1st} \\ 2 \text{ for 2nd} \\ 1 \text{ for 3rd} \\ 0 \text{ for other sentences} \end{cases}$$

On the other hand, considering that there are more than three paragraphs in the text.

$$\begin{cases} f2 = 1 \text{ if sentences occur in the first paragraph} \\ f2 = 0 \text{ if sentences occur in the middle paragraph} \\ f2 = 1 \text{ if sentences occur in the last paragraph} \end{cases}$$

3. **Term Weight**: The total term weight is calculated by computing tf and idf for document. Here idf refers to inverse document frequency which simply tells about whether the term is common or rare across all documents. The score of important score wi of word i can be calculated by the traditional $tf.idf$ methods. [20]

$$w_i = tf_i \cdot idf_i = tf_i \cdot \log \frac{N}{n_i}$$

$$\begin{cases} tf_i = \text{ the term frequency of word i in the document} \\ N = \text{ the total number of sentences} \\ n_i = \text{ the number of sentences in which word i occurs} \end{cases}$$

$$f3 = \frac{\sum_{i=1}^{k} w_i(S)}{Max(\sum_{i=1}^{k} w_i(S_i^N))}$$

$$k = \text{number of words in sentence}$$

Besides, there are other features generally used to calculate the scores of sentences including sentence length, proper nouns, thematic words and so on.

The third part is the summarization methods. Bayesian Classifier, neural networks and other machine learning-based techniques play a significant role in text summarization.

1. **Bayesian Classifier:** Given a training set of documents with hand-selected document extracts, developing a classification function that estimates the probability a given sentence is included in an extract. New extracts can then be generated by ranking sentences according to this probability and selecting a user-specified number of the top scoring ones. [21]
   For each sentence s, computing the probability it will be included in a summary S given the k features $f_j : j = 1...k$, which can be expressed by using Bayes' rule as follows:

$$P(s \in P|f1, f2, ..., fk) = \frac{P(f1, f2, ..., fk|s \in S)P(s \in S)}{P(f1, f2, ..., fk)}$$

   Assuming statistical independence of the features:

$$P(s \in P|f1, f2, ..., fk) = \frac{\prod_{j=1}^{k} P(f_j|s \in S)P(s \in S)}{\prod_{j=1}^{k} P(fj)}$$

2. **Neural Networks:**Artificial neural networks(ANN) are computational models inspired by the structure of neurons in the human brain. It consists of a large interconnected hierarchy of artificial neurons (or nodes), and the connections between these layers of neurons have weights that are learned to adjust based on input data. ANN can be used to learn complex patterns and features for classification, prediction, and decision making. [19] [21]
   The Neural Network has three layered feed-forward structure. It consists of n input layer neurons, $n - 1$ hidden layer neurons and one output layer neurons. Each sentence is represented as a vector $[f1, f2, ..., fn]$ which consists of n features. The features can be selected according to position of document, position of the sentence and other criteria mentioned above. [21]
   Text Summarization process inclues three phases: training, feature fusion and sentence selection. The first step is to train a neural network to recognize the type of sentences that should be added in the summary. In the second step, reducing the neural network and collapsing the hidden layer unit activations into discrete values with frequencies. The third step, sentence selection, using the modified neural network to filter the text and to select the highly ranked sentences only. [21]

2) Abstractive Summarization:

Abstractive summarization is another method of summary generation, which corresponds to extractive summarization. [22] In the process of generating an abstract, an abstract abstract does not simply extract a sentence or paragraph from the original text, but attempts to understand the text content and express the abstract in its own way, usually implemented using natural language processing techniques. The main idea of this approach is to use models to understand the meaning of text and reexpress the information in a new form or way. [22]

The methods for implementing abstract summarization usually involve the use of natural language processing (NLP) and machine learning techniques.

1. **RNN:** RNNS and their variants (such as long short-term memory network LSTM, gated loop unit GRU) are neural network models for processing sequence data, often used for generative tasks. These models can learn the semantics and structure of text sequences and generate summaries that are similar, but not identical, to the original text. [22]
2. **Encoder-decoder Architecture:** Similar to the encoder-decoder architecture in machine translation, the original text is encoded into an intermediate representation, which the decoder then converts into a summary. This architecture is often used to generate abstract abstracts, using encoders to understand text semantics and content, and decoders to generate new abstracts.
3. **Pre-trained Language Models:** Recent research on natural language processing has shown that pre-trained language models (such as BERT, GPT, etc.) can play an important role in abstract summary generation. Through large-scale text training, these models can understand the semantics of text and generate more logical and coherent summaries. [23]

3) Comparison between Extractive and Abstractive Summarization:

1. **Advantages and Disadvantages of Extractive Summarization:** Abstracting extracts sentences or paragraphs directly from the original text, thus preserving the accuracy and integrity of the original information. It is usually relatively simple and direct, the generation speed is fast, and the text does not need to be rebuilt and generated. However, only the existing information in the original text can be extracted and new expressions or content cannot be creatively generated. By extracting sentences or paragraphs directly, the resulting summarization may lack coherence and logic, affecting the reading experience. It is difficult to work with multiple source documents and to synthesize information to generate summarization. [21] [20]
2. **Advantages and Disadvantages of Abstractive Summarization:** The ability to reexpress information based on understanding the semantics and content of text, and the ability to generate content creatively. The resulting summaries are generally more fluid, coherent, and readable. Ability to process documents from multiple sources and combine information to generate summarization. However, due to the reexpression of content, inaccurate or incorrect information may be introduced, resulting in distorted information. Evaluating the quality of abstractive summarization is relatively difficult because the generation process is not as directly visible as extractive summarization. The computational complexity of generating abstractive summarization is usually higher than that of extractive summarization. [20]

B. *Multimodal and Multi-document Summarization*

1) Multimodal Summarization:

Multimodal Summarization involves generating summarization or generalizations from multiple modes of data, such as text, images, video, audio, etc. The goal is to combine multiple data sources to produce a comprehensive, comprehensive summarization that effectively summarizes the contents of the entire multimodal dataset. One of the challenges of multimodal summarization is to effectively merge information from different modes. In recent years, with the development of deep learning and multimodal learning, researchers have made some progress in the field of multimodal summarization.

1. **Models that Incorporate Multi-modal Features:** Using deep learning models such as Multimodal Recurrent Neural Networks (MRNN) or Multimodal Convolutional Neural Networks (MCNN). These models are capable of processing data from a variety of modes.
2. **Intermodal Alignment and Correspondence Learning:** Connect data from different modes through modal alignment and correspondence learning methods. Using techniques such as Multimodal Generative Adversarial Networks (M-GANs), information fusion between modes is achieved through adversarial training between generators and discriminators.
3. **Methods Based on Graph Structure:** Using a Graph structure model, data of different modes are represented as nodes in the graph, and multi-modal information is fused through techniques such as Graph Neural Networks. The graph-structured approach can effectively capture the relationships between different modal data and provide a way to fuse these relationships. [20]

These approaches all attempt to address the challenges of multimodal summarization generation to varying degrees, but the field remains challenging and more research is needed to improve the performance and effectiveness of models.

2) Multi-document Summarization:

Multi-document Summarization is the extraction of key information from multiple documents to produce a concise summarization containing the main content. Unlike a single document summarization, a multi-document summarization needs to process information from multiple documents in order to distil a general content. This method of summarization generation is usually used to process a large number of relevant documents, such as news reports, research papers, web articles and so on. It is designed to provide users with a concise summarization by summarizing and extracting key information from multiple documents, allowing them to quickly understand the overall content without having to read all the documents. [24]

There may be similar or duplicate content between multiple documents, and you need to avoid repeating the same information in the summarization. Choosing which information is critical, representative, and how to integrate it into a concise summarization is one of the challenges. When extracting information from multiple documents, it is important to maintain consistency and information integrity in the generated summarization.

1. **Extractive Methods**:
   - Sentence extraction based approaches: Extract the most representative or key sentences from multiple documents to form a summarization. This may involve techniques such as text similarity comparison, keyword extraction, and importance scoring. [24]
   - Graph model methods: The sentences or paragraphs of multiple documents are represented as graph structures, and then graph algorithms are used to identify and extract important information.
2. **Abstractive Methods**:
   - Generative models: Use generative models (such as RNN, LSTM, etc.) to understand multiple documents and reexpress and summarize the content in your own way to generate a new summarization.
   - Language generation models: Use pre-trained language generation models (such as GPT, BERT, etc.) to understand text and generate summarization that give a fuller understanding of semantics and context. [23]
3. **Clustering and Information Extractive Methods**:
   - Clustering methods: Cluster similar contents in multiple documents, and then extract representative information from each cluster as a summarization. [25]
   - Information Extractive Methods: Using information extractive techniques to extract facts, events, or key information from multiple documents and combining them into a summarization.
4. **Hybrid Methods and Integrated Models**:

- Hybrid Methods: Combining extractive and abstractive methods, leveraging the strengths of both methods to produce a more comprehensive and content-rich summarization.
- Integration Models: Integrate several different summary generation models together to produce a more accurate and comprehensive summary.

Each of these approaches has its advantages and disadvantages, and choosing the right one depends on the application requirements, document type, and data characteristics. Multi-document summarization generation is a complex task, which requires comprehensive consideration of information extraction, importance of content and coherence of summarization.

C. Evaluation and Factuality in Summarization Generation

1) Evaluation:

- Automatic Evaluation Metrics:
  1. **Precision and Recall:** They are two main criterion for evaluating the similarity between the summarization which is generated by the system and the one generated by human. [21]

$$Precision = \frac{Correct}{Correct + Wrong}$$

$$Recall = \frac{Correct}{Corrected + Missed}$$

     ”Correct” means the number of sentences that occur in both summaries which are generated by human and system. ”Wrong” means the number of sentences that occur in the summarization generated by system but not by human. ”Missed” means the number of sentences that occur in the summarization generated by human but not by system.
  2. **ROUGE:** There are several variations of the ROUGE metric, the most commonly used of which are ROUGE-N and ROUGE-L. [20] [26]
     ROUGE-N: Measuring n-gram overlap. It calculates the degree of n-gram (n consecutive words) overlap between the generated summarization and the reference summarization, usually unigram, bigram, or trigram. [20]
     ROUGE-L: Measuring the longest common subsequence. It takes into account the length of the longest common subsequence between the generated digest and the reference digest to evaluate the similarity between the two. [20]
  3. **BERTScore:** This index is mainly based on the BERT (Bidirectional Encoder Representations from Transformers) model, which uses the embedded representation of the BERT model to compare the similarity between the generated summarization and the reference text. [27] Cosine similarity is used as a measure of similarity. The advantage of BERTScore is that it takes into account the context of the context and is better able to capture sentence-level or even word-level similarity. [27]
- Human Evaluation
  1. **Intrinsic Assessment:** Human evaluators directly assess the quality, coherence, and informativeness of summarizations. [20]
  2. **User Studies:** Involves end-users reading and providing feedback on the usefulness and comprehensibility of the summarizations.

2) Factuality:

In summarization generation, factuality relates to the accuracy and factuality of the generated summarization content. This means whether the information contained in the summarization is true, accurate, and consistent with the original text or actual facts. It is important to keep the summarization accurate and factual, especially when the generated summarization is used to disseminate information or as a reference. When dealing with large amounts of information and content, it can be very challenging to ensure that the generated summarizations are consistent and accurate with the original information. Several methods can be used to improve the factual nature of the summarization, including [26]

1. **Information Verification and Fact Checking:** Fact-checking using reliable sources and data to ensure that the information in the summarization is accurate.
2. **Model Tuning and Training:** Adjust the summarization generation model to prioritize and generate more accurate and factual content. [26]
3. **Human Intervention and Editing:**After the summarization is generated, human editing and verification are performed to ensure that the generated content is accurate.

V. Dialogue Systems

A. An Antroduction to Dialogue System

Dialogue systems, also known as conversational agents or chatbots, represent a crucial component of human-computer interaction, aiming to facilitate natural and meaningful conversations between users and machines. These systems have evolved significantly, driven by advancements in natural language processing (NLP), machine learning, and artificial intelligence (AI). The primary goal of dialogue systems is to enable seamless communication between users and computers, allowing for information retrieval, task completion, as well as entertainment in a conversational manner.

Dialogue systems can be broadly categorized into two main types: task-oriented and chat-oriented. Task-oriented dialogue systems focus on accomplishing specific goals or tasks, such as booking a reservation, providing weather information, or assisting with customer support. On the other hand, chat-oriented dialogue systems prioritize engaging in open-ended and casual conversations, often for entertainment or companionship.

The technological underpinnings of dialogue systems include sophisticated NLP algorithms that enable the system to understand and generate human-like language. Early systems relied heavily on rule-based approaches, where predefined rules determined system responses. However, recent advancements have seen the integration of machine learning and deep learning techniques, allowing systems to learn from data and improve their performance over time.

The application domains of dialogue systems are diverse, ranging from virtual assistants on smartphones to customer support chatbots on websites. They play a crucial role in enhancing user experiences by providing efficient and user-friendly interactions. As dialogue systems continue to advance, researchers and developers are exploring ways to make these systems more context-aware, emotionally intelligent, and capable of handling complex and dynamic conversations.

B. A brief history of dialogue system

The early simple dialogue systems which appeared in the 1960s and 1970s were text-based. Generally speaking, EZILA was regarded as the first chatbot, who simulated a Rogerian psychotherapist [28], often in a convincing way, and has inspired many generations of chatbot authors for whom a major motivation is to develop a system that can pass Turing's Imitation Game [29]. Moreover, there was a dialogue system named BASEBALL, which operated as a question-answering system focused on baseball games, handling limited syntactic structures and rejecting questions it couldn't answer. Another one called SHRDLU was linguistically more advanced, incorporating a large grammar of English, semantic knowledge about objects in its domain (a blocks world), and a pragmatic component that processed non-linguistic information about the domain [30]. There was also a dialogue system designed for flight booking, GUS [31].

During the late 1970s and early 1980s, the researchers wants to go further, exploring topics like recognizing user intentions, cooperative behavior, and addressing miscommunications like misconceptions and false assumptions. In the plan-based model of dialogue, which gained prominence in the 1980s, speech acts like requests were formalized as action schemas akin to those utilized in AI planning models. This early work laid the foundation for subsequent theoretical developments in dialogue technology, including the BDI (Belief, Desire, Intention) model, Information State Update Theory, and Constructive Dialogue Modeling theory. However, a drawback of the plan-based approach was its computational complexity, potentially rendering it intractable in the worst-case scenarios.

In the late 1980s and early 1990s, the development of more powerful speech recognition engines led to the emergence of Spoken Dialogue Systems (SDSs). Examples include ATIS (Air Travel Information Service) in the U.S. and SUNDIAL, a European community-funded project. These systems, like MIT's Mercury, Ravenclaw and TRIPS, were often domain-specific, focusing on tasks such as flight inquiries. The DARPA Communicator systems explored multi-domain dialogues. However, speech recognition errors were common, leading to a focus on strategies for error detection and correction, including confirmation techniques. The MIT Mercury system interaction exemplifies challenges faced by these early SDSs.

Besides, researchers were also working hard on voice-based dialogue systems. Just before the year 2000, ATT's How May I Help You (HMIHY) systemwhich could support call routing by classifying cunstomer calls and route them to the correct destination, is an early example [32]. In the recent years, the voice-based dialogue systems also have developed a lot, including design and evaluation guidelines, development of standards, improving toolkits, speech analytics and usability testing.

With the development of different aspects of NLP technology, including neural network, deep learning, the dialogue systems also grew into new versions. At the same time, dialogue also developed a lot by combining other technics. For example, more detailed interactions by adding virtual animated character to the chatbot [33].

Recently, ChatGPT have already became the most famous Chatbot. Large-Scale Pretrained Models has provided new momentum for the development of dialogue systems. These models, pretrained on massive text corpora, can learn richer language representations, enhancing the performance of dialogue systems. Furthermore, researches about applications of those well-performed LLMs are also carried out. For example, the researcher from MIT designed a way to apply chatGPT in order to generate human agent, which can be used in games and social or mental experiments [34].

In this rapidly evolving field, the development of dialogue systems holds the promise of creating more intuitive and human-like interactions between individuals and machines, transforming the way we communicate with technology in various aspects of our daily lives.

C. Bias Handling and Factuality Assurance in Dialogue Systems

1) Bias Handling:

1. **Diverse and Inclusive Data**: To mitigate bias, it's essential to use diverse and inclusive training datasets. These datasets should represent a wide range of demographics, dialects, and cultural contexts to ensure the chatbot does not favor or discriminate against any particular group.
2. **Algorithmic Fairness**: Employing algorithms that actively detect and reduce bias is crucial. This can involve techniques like fairness-aware modeling, which adjusts the algorithm to treat different groups equitably.
3. **Continuous Monitoring and Updating**: Post-deployment, continuous monitoring of the chatbot's responses is vital to identify and rectify any emerging biases. Regular updates based on diverse user feedback can help maintain an unbiased dialogue system. [35]

2) Factuality Assurance:

1. **Robust Fact-Checking Mechanisms**: Implementing robust fact-checking mechanisms within the system ensures the accuracy of the information provided. This could involve cross-referencing answers with reliable and up-to-date data sources. [36]
2. **Limiting Speculation**: Dialogue systems should be designed to avoid speculation or assumption-based responses. If uncertain, the system should acknowledge its limitations and, where possible, refer users to human experts or additional resources.

D. LLMs in Dialogue Systems

The connection between chatbots and Large Language Models (LLMs) like GPT-3 or BERT is significant. LLMs have transformed chatbot technology by enabling a deeper understanding of natural language. These models, trained on vast amounts of text data, can understand and generate human-like responses, making chatbot interactions more fluid, context-aware, and responsive to the nuances of human language. The integration of LLMs into chatbots marks a pivotal advancement in the field, allowing for more sophisticated, helpful, and engaging conversational agents in various applications, from customer service to personal assistants.

1) trafficGPT: **Introduction:** TrafficGPT represents an innovative integration of ChatGPT with traffic-specific models, targeting the complexities of urban traffic management. LLMs exhibit advanced reasoning and planning capabilities.

TrafficGPT System: TrafficGPT extends ChatGPT's functionality, equipping it with langchain to analyze and process traffic data, thereby offering crucial decision-making support for transportation systems. This system adopts a structured methodology to intelligently breakdown and execute complex traffic-related tasks using dedicated traffic foundation models.

Key Enhancements: The core enhancements of TrafficGPT include: 1) Augmenting ChatGPT's capabilities to process traffic data and assist in decision-making; 2) Streamlining the decomposition of intricate tasks and sequential deployment of traffic models; 3) Supporting human decision-making in traffic control via natural language interactions; 4) Providing a platform for interactive feedback and outcome modifications.

Conclusion: In conclusion, TrafficGPT serves as a bridge between the realms of LLMs and traffic-specific models, marking a significant advancement in traffic management. It showcases the transformative potential of AI in revolutionizing both the analysis and decision-making processes in traffic management. [37]

2) AVA: A Financial Service Chatbot Based on Deep Bidirectional Transformers: BERT (Bidirectional Encoder Representations from Transformers), developed by Google, marks a transformative advancement in natural language processing (NLP). Its groundbreaking feature is bidirectional training, which comprehensively analyzes the context of a word from both its preceding and following text, a notable departure from earlier unidirectional models. This method yields a more nuanced understanding of language, profoundly enhancing tasks like text classification, question answering, and language translation. BERT's adoption across various NLP applications has led to remarkable improvements in their performance, showcasing its effectiveness in comprehending and generating human language.

In a recent study, researchers developed AVA, a financial service chatbot, using the BERT model to refine customer interactions. [38] AVA stands out in its ability to discern a broad spectrum of customer intents and skillfully determines when to escalate complex inquiries to human agents. A key innovation of this study is the integration of BERT for estimating uncertainties in intent prediction, marking a significant leap in chatbot technologies. Additionally, the researchers creatively employed BERT for automated spelling correction, boosting AVA's comprehension of customer queries. Designed for in-house deployment using open-source tools, AVA demonstrates its practicality in fields like finance that handle sensitive data, illustrating the evolving role of conversational AI in enhancing chatbot functionalities.

E. Exploring the llms ability behind chatbot

1) llmtime:

This technique, termed LLMTIME, encodes time series data as strings of numerical digits and interprets forecasting as next-token prediction in text, a fundamental mechanism in LLMs. This approach enables LLMTIME to extrapolate time series data in a zero-shot manner, which means it requires no specific training on the target dataset. It not only matches but, in some cases, exceeds the performance of dedicated time series models. [39]

LLMTIME's remarkable performance is attributed to the inherent qualities of LLMs, such as their ability to handle simple or repetitive sequences, which align well with the typical structure of time series data like seasonality. Additionally, LLMs can efficiently manage missing data and interpret multimodal distributions, both critical aspects in time series analysis. The method can also incorporate additional side information and explain its predictions, enhancing its utility and interpretability. Interestingly, the performance of LLMTIME scales with the power of the underlying base model, although it's noted that GPT-4 sometimes underperforms compared to GPT-3 due to differences in tokenization and uncertainty calibration.

2) AuxMobLCast:

This paper proposes a unique pipeline named AuxMobLCast, which employs existing pre-trained language models to forecast human mobility by converting numerical temporal sequences into natural language sentences through specific prompts. The primary goal is to bridge the gap between traditional numerical forecasting methods and the expressive capabilities of natural language processing models. [40]

AuxMobLCast Pipeline: The AuxMobLCast pipeline integrates an auxiliary Point-of-Interest (POI) category classification task with an encoder-decoder architecture. This design is motivated by the correlation between POI categories and their visiting patterns, which is critical for accurately forecasting visitor flows at various locations.

Transforming Mobility Data: The process involves transforming human mobility data into natural language sentences, enabling the direct application of pre-trained language models in fine-tuning phases for mobility prediction. This method not only predicts numerical values but also incorporates contextual and semantic information, providing a more holistic approach to forecasting.

Experimental Setup and Evaluation: The study utilized real-world human mobility data from three major cities, applying the pre-train and fine-tune paradigm with pre-trained language models sourced from HuggingFace. The AuxMobLCast pipeline was fine-tuned using generated mobility prompts, and the models' performance was evaluated using metrics like Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

Performance Insights: When comparing different encoder models, it was found that BERT as an encoder in the AuxMobLCast pipeline outperformed other models like Informer and Transformer-B on average. This was particularly evident in the dataset from Dallas, suggesting the versatility and effectiveness of BERT in handling complex forecasting tasks.

Potential for Future Research: The research presents pioneering work in applying pre-trained language foundation models for sequential temporal data forecasting. The results indicate that such an approach can be a new direction in addressing human mobility forecasting tasks, with significant implications for smart city planning and transportation management.

VI. Summary

This comprehensive review charts the evolution and current state of Natural Language Processing (NLP), discussing core areas like coreference resolution, discourse parsing, and summarization generation. It delves into challenges in NLP, including bias handling and factuality in dialogue systems. Key advancements like the integration of Large Language Models (LLMs) and their impact across various NLP applications are examined. The report underscores the growing complexity and potential of NLP, reflecting its critical role in advancing machine understanding of human language and broadening its applicability in numerous fields.

References

[1] K. S. Jones, "Natural language processing: a historical review," *Current issues in computational linguistics: in honour of Don Walker*, pp. 3–16, 1994.

[2] P. S. Jacobs, Ed., *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval*. USA: L. Erlbaum Associates Inc., 1992.

[3] C. J. Weinstein, "Overview of the 1994 arpa human language technology workshop," in *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994.

[4] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Interspeech*, vol. 2, no. 3. Makuhari, 2010, pp. 1045–1048.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[8] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[9] R. Sukthanker, S. Poria, E. Cambria, and R. Thirunavukarasu, "Anaphora and coreference resolution: A review," *Information Fusion*, vol. 59, pp. 139–162, 2020.

[10] I. Porada, A. Olteanu, K. Suleman, A. Trischler, and J. C. K. Cheung, "Investigating failures to generalize for coreference resolution models," *arXiv preprint arXiv:2303.09092*, 2023.

[11] A. Marasović, L. Born, J. Opitz, and A. Frank, "A mention-ranking model for abstract anaphora resolution," *arXiv preprint arXiv:1706.02256*, 2017.

[12] A. Cattan, A. Eirew, G. Stanovsky, M. Joshi, and I. Dagan, "Realistic evaluation principles for cross-document coreference resolution," *arXiv preprint arXiv:2106.04192*, 2021.

[13] S. Joty, G. Carenini, R. Ng, and G. Murray, "Discourse analysis and its applications," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, 2019, pp. 12–17.

[14] J. Li, M. Liu, B. Qin, and T. Liu, "A survey of discourse parsing," *Frontiers of Computer Science*, vol. 16, no. 5, p. 165329, 2022.

[15] L. Carlson, D. Marcu, and M. E. Okurowski, "Building a discourse-tagged corpus in the framework of rhetorical structure theory," *Current and new directions in discourse and dialogue*, pp. 85–112, 2003.

[16] S. Hou, S. Zhang, and C. Fei, "Rhetorical structure theory: A comprehensive review of theory, parsing methods and applications," *Expert Systems with Applications*, vol. 157, p. 113421, 2020.

[17] R. Prasad, B. Webber, and A. Lee, "Discourse annotation in the pdtb: The next generation," in *Proceedings 14th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, 2018, pp. 87–97.

[18] H. Chen, X. Liu, D. Yin, and J. Tang, "A survey on dialogue systems: Recent advances and new frontiers," *Acm Sigkdd Explorations Newsletter*, vol. 19, no. 2, pp. 25–35, 2017.

[19] S. Kavyashree, R. Sumukha, R. Soujanya, and S. Tejaswini, "Survey on automatic text summarization using nlp and deep learning," in *2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS)*. IEEE, 2023, pp. 523–527.

[20] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "Text summarization techniques: a brief survey," *arXiv preprint arXiv:1707.02268*, 2017.

[21] O. Tas and F. Kiyani, "A survey automatic text summarization," *PressAcademia Procedia*, vol. 5, no. 1, pp. 205–213, 2007.

[22] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[23] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–40, 2023.

[24] J. Goldstein, V. O. Mittal, J. G. Carbonell, and M. Kantrowitz, "Multi-document summarization by sentence extraction," in *NAACL-ANLP 2000 workshop: automatic summarization*, 2000.

[25] Z. Cao, F. Wei, L. Dong, S. Li, and M. Zhou, "Ranking with recursive neural networks and its application to multi-document summarization," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.

[26] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On faithfulness and factuality in abstractive summarization," *arXiv preprint arXiv:2005.00661*, 2020.

[27] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[28] J. Weizenbaum, "Eliza—a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[29] A. M. Turing, *Computing machinery and intelligence*. Springer, 2009.

[30] M. McTear, *Conversational ai: Dialogue systems, conversational agents, and chatbots*. Springer Nature, 2022.

[31] D. G. Bobrow, R. M. Kaplan, M. Kay, D. A. Norman, H. Thompson, and T. Winograd, "Gus, a frame-driven dialog system," *Artificial intelligence*, vol. 8, no. 2, pp. 155–173, 1977.

[32] A. L. Gorin, G. Riccardi, and J. H. Wright, "How may i help you?" *Speech communication*, vol. 23, no. 1-2, pp. 113–127, 1997.

[33] J. Cassell, "Embodied conversational agents: representation and intelligence in user interfaces," *AI magazine*, vol. 22, no. 4, pp. 67–67, 2001.

[34] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023, pp. 1–22.

[35] E. Sheng, J. Arnold, Z. Yu, K.-W. Chang, and N. Peng, "Revealing persona biases in dialogue systems," *arXiv preprint arXiv:2104.08728*, 2021.

[36] P. Gupta, C.-S. Wu, W. Liu, and C. Xiong, "Dialfact: A benchmark for fact-checking in dialogue," *arXiv preprint arXiv:2110.08222*, 2021.

[37] S. Zhang, D. Fu, Z. Zhang, B. Yu, and P. Cai, "Trafficgpt: Viewing, processing and interacting with traffic foundation models," *arXiv preprint arXiv:2309.06719*, 2023.

[38] S. Yu, Y. Chen, and H. Zaidi, "Ava: A financial service chatbot based on deep bidirectional transformers," *Frontiers in Applied Mathematics and Statistics*, vol. 7, p. 604842, 2021.

[39] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson, "Large language models are zero-shot time series forecasters," *arXiv preprint arXiv:2310.07820*, 2023.

[40] H. Xue, B. P. Voutharoja, and F. D. Salim, "Leveraging language foundation models for human mobility forecasting," in *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, 2022, pp. 1–9.