Soyeon Caren Han · Henry Weld ·
Yan Li · Jean Lee · Josiah Poon

Natural Language Understanding in Conversational AI with Deep Learning

Springer

Natural Language Understanding in Conversational AI with Deep Learning

Soyeon Caren Han • Henry Weld • Yan Li •
Jean Lee • Josiah Poon

Natural Language Understanding in Conversational AI with Deep Learning

Springer

Soyeon Caren Han
School of Computing and Information Systems
The University of Melbourne
Melbourne, VIC, Australia

Yan Li
School of Computer Science
The University of Sydney
Sydney, NSW, Australia

Josiah Poon
School of Computer Science
The University of Sydney
Sydney, NSW, Australia

Henry Weld
School of Computer Science
The University of Sydney
Sydney, NSW, Australia

Jean Lee
School of Computer Science
The University of Sydney
Sydney, NSW, Australia

ISBN 978-3-031-74363-4 ISBN 978-3-031-74364-1 (eBook)
https://doi.org/10.1007/978-3-031-74364-1

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Language has always been a crucial part of being human. From the basic sounds our ancestors made to the detailed conversations we have today, language helps us connect and understand each other. It goes beyond just talking; it is the foundation of our culture and civilization. However, teaching machines to understand human language is a big challenge. This book explores the journey of Natural Language Understanding (NLU), showing how we have gone from simple human interactions to advanced computer-human conversations.

Natural Language Understanding is about bridging the gap between human language and computers. It involves making machines understand, interpret, and respond to human language accurately and in context. NLU is essential in today's world, from virtual assistants that help us daily to AI systems driving innovation in many fields.

The development of NLU has been amazing. At first, researchers used rule-based systems to create language rules. These early systems, like ELIZA and SHRDLU, were groundbreaking but struggled with the complexity of natural language. Creating rules for every possible scenario was tough and often impractical.

Then came statistical methods. Researchers started using statistical models to understand the probabilities in language, moving away from rigid rules. This opened new ways to handle language tasks like text generation and classification. Techniques like n-gram models and Hidden Markov Models helped, but they still had limitations, especially with long-range dependencies and rare events.

Machine learning took NLU further. Algorithms like decision trees, support vector machines, and logistic regression were used for language tasks. These methods allowed computers to learn from data, leading to applications like sentiment analysis and machine translation. But the real breakthrough came with deep learning. Deep learning has changed NLU by using neural networks similar to the human brain. These networks, like recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformers, have greatly improved language processing. Innovations like word embeddings and models like BERT and GPT have enhanced our understanding of language even more.

One key lesson from NLU's progress is that understanding language depends on the situation. Effective communication needs commonsense reasoning and context. Large language models like GPT-3 and GPT-4 are powerful but still struggle with context and nuance, showing the need for better NLU techniques.

NLU can be used in many ways. From improving customer service with chatbots to enhancing accessibility with speech recognition, NLU technologies are transforming many areas. But there are challenges too. NLU systems need to be context-aware, culturally sensitive, and capable of nuanced understanding.

This book will guide you through the history, current advancements, and future of NLU.

Chapter 1: Introduction to Natural Language Understanding Chapter 1 lays the foundation by tracing the evolution of NLU from early human communication to modern computer-human interactions. It underscores the significance of language in human society and the challenges in making machines understand and respond to human language.

Chapter 2: Prerequisites and Glossary for Natural Language Understanding Chapter 2 serves as a foundational resource, consolidating essential prerequisites and key terminologies drawn from across the book's chapters. This chapter aims to provide clarity and insight into the foundational principles underlying NLU and dialogue management.

Chapter 3: Single-Turn Natural Language Understanding Chapter 3 looks at Single-Turn NLU, focusing on tasks that involve interpreting and processing user inputs in a single interaction. It covers essential tasks like Intent Classification and Slot Filling, exploring both traditional and modern machine learning approaches. The chapter also examines the role of external resources and key datasets in enhancing NLU systems.

Chapter 4: Multi-Turn Natural Language Understanding Chapter 4 moves to Multi-Turn NLU, where dialogue systems have extended interactions with users. It explores techniques for managing dialogues, using context, and integrating external knowledge bases. Prominent datasets and methods for handling multi-turn conversations are also discussed.

Chapter 5: Evaluating Natural Language Understanding Chapter 5 focuses on the methods for evaluating NLU systems. It discusses the annotation of datasets and various performance assessment methods, covering different levels of understanding, from intent recognition to slot filling and domain classification.

Chapter 6: Applications and Case Studies in Natural Language Understanding Chapter 6 shows practical applications of NLU in various fields, including finance, medicine, and law. It features case studies that highlight the impact of NLU technologies in real-world scenarios.

Chapter 7: Challenge, Conclusion, and Future Direction Chapter 7 explores the core obstacles hindering the advancement of NLU, including ambiguity, domain adaptation, data scarcity, and ethical concerns. By understanding these challenges, this final chapter highlights the ongoing work needed to advance NLU.

This book aims to inspire and equip everyone interested in NLU with the knowledge and tools to advance this field. Join us to explore the details of Natural Language Understanding and see its profound impact on our past, present, and future.

Welcome to the exciting world of Natural Language Understanding!

Melbourne, VIC, Australia — Soyeon Caren Han
Sydney, NSW, Australia — Henry Weld
Sydney, NSW, Australia — Yan Li
Sydney, NSW, Australia — Jean Lee
Sydney, NSW, Australia — Josiah Poon
Aug 2024

Contents

Chapter 1
Introduction to Natural Language Understanding

1.1 Overview

In the vast tapestry of human civilization, one thread binds us all together, the language. From the primal grunts of our ancestors to the eloquent prose of modern literature, language has been the cornerstone of human communication and understanding. Yet, for machines, deciphering the intricate nuances of language has long been an enigma.

Welcome to the exploration of Natural Language Understanding (NLU), a journey that spans centuries of human ingenuity, technological innovation, and linguistic curiosity. In this book, we embark on a quest to unravel the mysteries of how machines have endeavored to comprehend the language that defines our existence.

NLU represents humanity's quest to bridge the gap between human language and computational systems. It encompasses the formidable challenge of enabling machines to comprehend, interpret, and respond to human language in a manner that is not only accurate but also contextually aware and meaningful.

The significance of NLU permeates every facet of modern society, from powering virtual assistants that help us navigate our daily lives to driving advancements in artificial intelligence that revolutionize industries. Whether we realize it or not, NLU plays a pivotal role in shaping the technological landscape of our world.

This book embarks on a journey through the annals of time, tracing the evolution of NLU from its humble beginnings to its current state of sophistication. We will explore the ingenuity of early pioneers who dared to dream of machines that could understand human language, the watershed moments that marked paradigm shifts in NLU methodologies, and the relentless pursuit of innovation that propels us toward a future where seamless human-machine communication is not just a possibility but a reality. We briefly delve into the earliest attempts to automate

S. C. Han et al., *Natural Language Understanding in Conversational AI with Deep Learning*, https://doi.org/10.1007/978-3-031-74364-1_1

language understanding and explore the transformative impact of machine learning and neural networks. Along the way, we encounter the practical applications that have reshaped industries and revolutionized human-machine interaction.

Through the pages that follow, we invite you to delve into the captivating narrative of how humanity's quest to understand language has intertwined with the relentless march of technological progress. Join us as we unravel the mysteries of Natural Language Understanding and discover the profound impact it has had on our past, present, and future.

Welcome to the fascinating world of Natural Language Understanding!!

1.2 A Brief History of Natural Language Understanding (NLU)

Natural Language Understanding (NLU) has been a pursuit since the early days of computing. While the concept may seem straightforward to humans, teaching computers to understand human language presented significant challenges that early pioneers bravely tackled.

1.2.1 Early Rule-Based NLU

In the nascent stages of NLU, researchers focused on rule-based systems. These systems attempted to codify language understanding through explicit sets of rules. One of the earliest notable efforts was by Alan Turing, who proposed the Turing Test in 1950 as a benchmark for gauging a machine's ability to exhibit intelligent behavior indistinguishable from that of a human.

A milestone in NLU came with Joseph Weizenbaum's introduction of **ELIZA** [230] in the mid-1960s. Although primitive by today's standards, ELIZA simulated conversation by recognizing keywords in a user's input and generating pre-scripted responses. Despite its simplicity, ELIZA sparked interest in human-computer interaction and laid the groundwork for future chatbot development (Fig. 1.1).

In the late 1960s, Terry Winograd developed **SHRDLU** [239], a program capable of understanding and executing commands in a simulated world of blocks. SHRDLU demonstrated the power of symbolic reasoning in NLU, as it manipulated symbols representing objects and actions according to predefined rules. While limited in scope, SHRDLU showcased the potential of symbolic approaches to language understanding (Fig. 1.2).

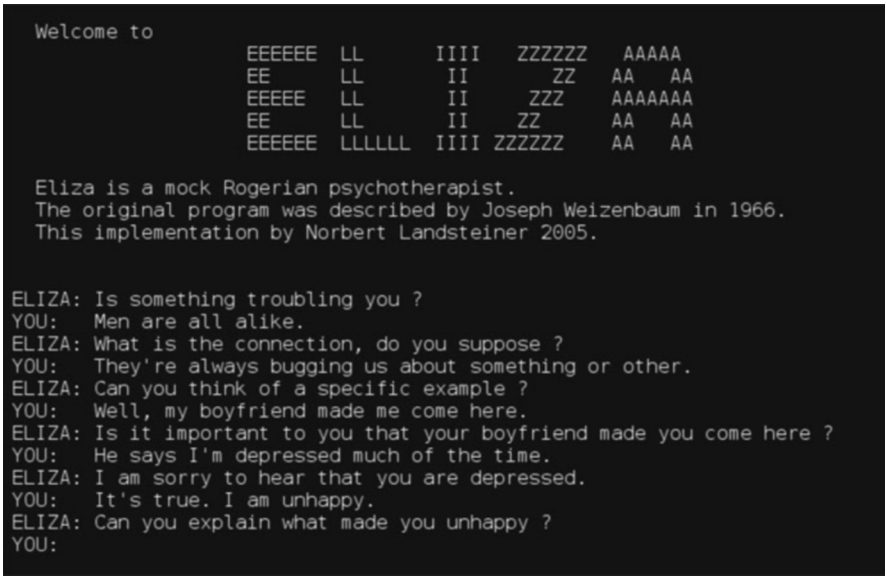

Fig. 1.1 A sample screenshot from a conversation with ELIZA [230], which recognizes the keywords by using predefined patterns and rules

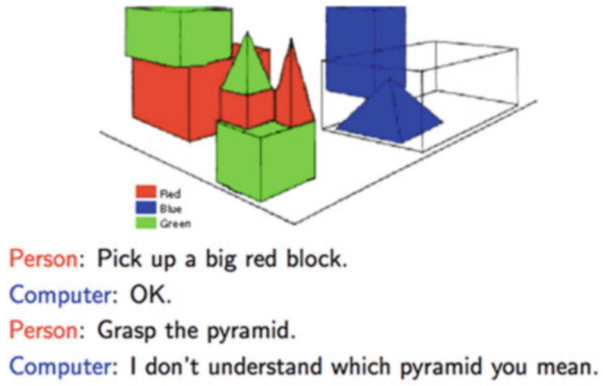

Fig. 1.2 A sample screenshot from a conversation with SHRDLU [239], which recognizes the keywords from the user's query by using a parser to analyze the syntactic structure

Early rule-based systems faced several challenges. They struggled to handle the complexity and ambiguity inherent in natural language. Crafting exhaustive rule sets to cover all possible linguistic nuances proved daunting and often impractical. Additionally, these systems lacked the ability to learn from data, relying instead on handcrafted rules that required constant maintenance and updating.

Despite their limitations, early attempts at NLU laid the groundwork for future advancements. They introduced foundational concepts and inspired researchers to

explore new approaches. The experiences gained from working with rule-based systems highlighted the need for more flexible and adaptive methods, leading to the emergence of statistical and, later, machine learning-based approaches to NLU.

In retrospect, these early endeavors serve as important milestones in the ongoing quest to bridge the gap between human language and machine understanding. While the journey was fraught with challenges, the lessons learned paved the way for the remarkable progress that followed in the field of natural language processing.

1.2.2 Statistical Approach-Based NLU

In the mid-twentieth century, there was a notable shift toward statistical approaches in NLU. Rather than relying solely on handcrafted rules, researchers began to leverage statistical models to capture the probabilistic nature of language. This transition marked a significant departure from the deterministic nature of rule-based systems and opened new avenues for tackling the complexities of natural language.

One of the earliest statistical models used in NLU is the n-gram model. N-grams are sequences of n words or characters that are used to model the probability of a word occurring given its context. By analyzing the frequency of n-grams in a corpus of text, language models can predict the likelihood of different word sequences, enabling tasks such as language generation and text classification. **Hidden Markov Models (HMMs)** [173] also played a significant role in statistical approaches to NLU. HMMs are probabilistic models that represent sequences of observable events generated by underlying hidden states. In the context of language processing, HMMs can be used for tasks such as speech recognition, part-of-speech tagging, and named entity recognition by modelling the sequential nature of language.

The rise of machine learning further propelled the adoption of statistical methods in NLU. Researchers began applying algorithms such as decision trees [148], support vector machines (SVMs) [39], and logistic regression [42] to various language processing tasks. These algorithms allowed computers to learn patterns and associations from labelled data, enabling tasks like sentiment analysis, text categorization, and machine translation.

While statistical approaches represented a significant advancement in NLU, they were not without challenges. These methods often struggled with capturing long-range dependencies in language, as well as handling rare or unseen events. Additionally, statistical models require large amounts of annotated data for training, which could be costly and time-consuming to acquire.

1.2.3 Rise of Machine Learning-Based NLU

In the latter half of the twentieth century, researchers increasingly turned to machine learning methods for NLU tasks due to the advances of hardware performance and efficiency. Unlike rule-based systems and early statistical approaches, machine learning algorithms could automatically discover patterns in data, reducing the need for handcrafted rules and heuristics.

Various machine learning algorithms were applied to NLU tasks, including **decision tree** [148], **support vector machines (SVMs)** [39], and **logistic regression** [42]. A key advantage of machine learning in NLU was its data-driven approach. By training models on large amounts of labelled data, researchers could capture the complex patterns and structures inherent in natural language. This data-driven paradigm enabled computers to generalize from examples and adapt to new linguistic phenomena.

One notable example of a traditional machine learning-based NLU system is IBM Watson [55]. **IBM Watson** NLU utilizes a combination of traditional machine learning algorithms and rule-based techniques to analyze and understand text data (Fig. 1.3).

IBM Watson initially processes unstructured text data, such as documents, social media posts, or customer feedback, to extract relevant information and insights. The system employs various feature extraction techniques to transform raw text into numerical representations that machine learning algorithms can process. These features include word frequencies, part-of-speech tags, syntactic patterns, and semantic representations. Watson employs a range of traditional machine learning

Fig. 1.3 A snapshot of the Jeopardy! challenge when Watson competes against humans and wins the round

algorithms, including decision trees, support vector machines (SVM), and logistic regression, to perform tasks such as sentiment analysis, entity recognition, and topic classification. It also incorporates rule-based processing to handle specific linguistic phenomena or domain-specific rules. These rules help enhance the accuracy and robustness of the system's NLU capabilities.

1.2.4 Deep Learning-Based NLU

Deep learning, a subfield of machine learning, has emerged as a powerful paradigm for natural language understanding (NLU). Deep neural networks, inspired by the structure and function of the human brain, have revolutionized language processing by enabling computers to learn hierarchical representations of text data.

Neural networks, originally proposed in the 1940s, experienced a resurgence in interest in the twenty-first century due to advances in computational power and data availability. Various neural network architectures, including recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer models, have been developed for NLU tasks.

One of the key innovations in deep learning for NLU is the concept of word embeddings. Word embeddings are dense vector representations of words learned from large text corpora. These representations capture semantic relationships between words, enabling models to better understand and manipulate language.

Recent advancements in deep learning have focused on transfer learning and pre-trained models. Contextualized word representations, exemplified by models like **BERT (Bidirectional Encoder Representations from Transformers)** [48] and **GPT (Generative Pre-trained Transformer)** [174], have further improved semantic understanding in NLU. These models capture the context in which words appear in a sentence, leading to more nuanced and contextually appropriate representations of meaning.

One prominent example of a Deep Learning-based Natural Language Understanding (NLU) system is OpenAI's GPT (Generative Pre-trained Transformer) series of models. GPT models, including **GPT-2** [175] and **GPT-3** [21], represent a significant advancement in NLU using large-scale deep learning architectures, specifically transformer models. GPT models are based on the transformer architecture, which allows them to capture long-range dependencies in text data. This architecture consists of multiple layers of self-attention mechanisms, enabling the model to weigh the importance of different words in a sentence dynamically. During pre-training, the model learns to predict the next word in a sequence given the preceding context, thereby capturing rich semantic and syntactic information from the data. After pre-training, GPT models can be fine-tuned on specific NLU tasks using supervised learning techniques. One key feature of GPT models is their ability to generate contextualized word representations. These representations capture the

meaning of words in context, allowing the model to produce more accurate and contextually appropriate responses to queries.

By pre-training large neural network models on massive text corpora, researchers have created models that can be fine-tuned on specific NLU tasks with relatively little labelled data, leading to improved performance and efficiency. The NLU is now likely to be shaped by ongoing advances in deep learning and its integration with other fields such as reinforcement learning and multimodal learning.

The history of natural language understanding (NLU) is a testament to human ingenuity and perseverance in bridging the gap between human language and machine understanding. From early rule-based systems to the rise of machine learning and deep learning, researchers have continuously pushed the boundaries of what computers can achieve in understanding and processing natural language.

Throughout this journey, we have witnessed remarkable advancements in NLU, driven by a combination of theoretical insights, algorithmic innovations, and exponential growth in computational resources and data availability. These advancements have led to transformative applications in areas such as virtual assistants, chatbots, sentiment analysis, and machine translation, reshaping the way we interact with technology and each other.

1.3 Unveiling the Limitations of Large Language Models in NLU

In the ever-evolving landscape of Natural Language Understanding (NLU), the advent of Pre-trained Language Models (PLMs) and their larger counterparts, Large Language Models (LLMs), has sparked a revolution in conversational systems. These models capture audiences with their uncanny ability to engage in seemingly human-like dialogue.

Yet, amid the marvel of their performance, there exists a paradox: the belief that these sophisticated models have rendered further advancements in NLU redundant. However, beneath the surface lies a labyrinth of challenges haunting LLM-based NLU. Despite their prowess, these models often stumble when it comes to grasping the intricate nuances of context and identifying pivotal information, occasionally conjuring perplexing or downright nonsensical responses. These shortcomings serve as a stark reminder of the ongoing need for innovative NLU techniques that can adeptly navigate the complexities of human language and cognition.

Let's go through some qualitative and quantitative examples together!

1.3.1 Qualitative Analysis

For a qualitative example illustrated in Fig. 1.4, consider a scenario where an LLM is tasked with solving the classic puzzle: *"Move the wolf, sheep, and cabbage to the opposite shore using the boat. Each time, the boat can only take three items, so multiple crossings are possible. Be careful; when the man is not around, the wolf will eat the sheep, and the sheep will eat cabbage."* When ChatGPT-4 [6] attempts to solve this, it erroneously selects just one item (the sheep) and takes it across the river, despite the instruction that three items can be taken at once. Even more intriguingly, when the user reiterates, "I said you can take three items," ChatGPT-4 initially makes the man take three items but then illogically returns with two items.

Furthermore, when the user queries, *"Can you find the value of intent and slot tokens from the user's query?"* ChatGPT-4 identifies the intent as solving the river crossing puzzle and recognizes the slot tokens as the items (wolf, sheep, and cabbage) and the transport (boat). However, it fails to grasp the crucial information that "three items" can be taken, a key detail necessary for correctly answering the question (Fig. 1.5).

These errors made by ChatGPT4 occur from several fundamental limitations inherent in Large Language Models (LLMs):.

- **Contextual Understanding:** LLMs, including ChatGPT, often struggle with maintaining complex contexts, especially when multiple steps or conditions are involved. If the conversation requires a nuanced understanding of constraints, it can be challenging for LLMs to navigate the key information.
- **State Management:** LLM models normally do not have a persistent memory or state-tracking mechanism. They process each query in isolation to some extent, which can lead to inconsistencies when handling multi-step problems that require remembering previous actions and their outcomes.
- **Training Data Limitations:** The models are trained on vast amounts of text from the Internet. However, this training does not always equip them to handle specific logical puzzles or niche scenarios that they have not encountered frequently during training.
- **Reasoning:** While LLMs are good at generating plausible-sounding text, they are not as adept at reasoning through problems in a logically consistent manner. Tasks requiring step-by-step reasoning, such as the river crossing puzzle, often expose these weaknesses.

LLMs primarily rely on pattern recognition rather than a deep understanding of underlying concepts. They might recognize common patterns in text related to puzzles but fail to internalize the logic needed to solve them accurately.

These limitations illustrate why LLMs, despite their impressive capabilities, cannot fully replace more sophisticated Natural Language Understanding techniques

Move the wolf, sheep and cabbage to the opposite shore using the boat. Each time the boat can only take three items and so multiple crosses is possible. Be careful, when the man is not around, the wolf will eat the sheep, and the sheep will eat cabbage.

To solve this classic puzzle, you need to transport the wolf, sheep, and cabbage across the river without the wolf eating the sheep or the sheep eating the cabbage. Here's a step-by-step solution:

1. The man takes the sheep across the river and leaves it on the opposite shore.
2. The man returns alone.
3. The man takes the wolf across the river.
4. The man leaves the wolf on the opposite shore and takes the sheep back with him.
5. The man leaves the sheep on the original shore and takes the cabbage across the river.
6. The man leaves the cabbage with the wolf on the opposite shore.
7. The man returns alone.
8. Finally, the man takes the sheep across the river.

Now all items are safely on the opposite shore without the wolf eating the sheep or the sheep eating the cabbage.

I said you can take three items

Apologies for that oversight. Let's revise the solution to accommodate the constraint of taking only three items at a time:

1. The man takes the wolf, sheep, and cabbage across the river.
2. The man returns alone.
3. The man takes the sheep and cabbage across the river.
4. The man leaves the sheep and cabbage on the opposite shore and takes the wolf back with him.
5. The man leaves the wolf on the original shore and takes the sheep and cabbage across the river.
6. The man leaves the sheep and cabbage on the opposite shore.
7. The man returns alone.
8. Finally, the man takes the wolf across the river.

Now, all items are safely on the opposite shore without the wolf eating the sheep or the sheep

Fig. 1.4 A snapshot of the limitation of LLM (ChatGPT4—June 9, 2024)

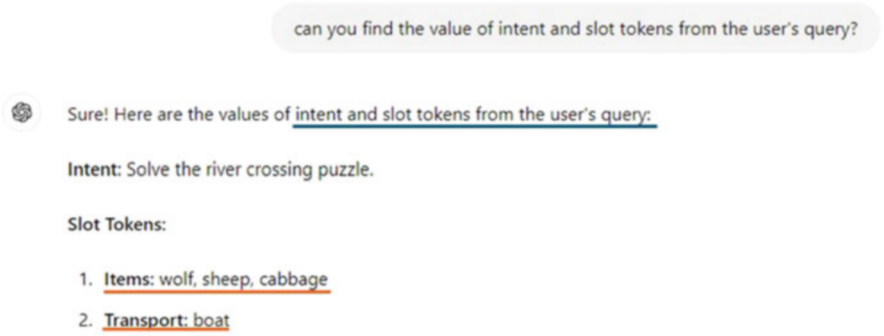

Fig. 1.5 A snapshot of the limitation of LLM (ChatGPT4—June 9, 2024)

Table 1.1 Evaluation on MultiWoz (MWOZ) and M2M datasets

	Intent detection (ID)		Slot filling (SF)	
	MWOZ (Acc.)	M2M (Acc.)	MWOZ (F1)	M2M (F1)
LLaMa2-7b-chat	0.4751	0.3363	0.0217	0.0751
LLaMa2-13b-chat	0.1679	0.2013	0.0891	0.1092
LLaMa2-70b-chat	0.3896	0.3275	0.0619	0.0883
Gemma-7b	0.6515	0.4588	0.6653	0.4357
GPT3.5	0.6971	0.5100	0.8175	0.5516
Our best model	**0.8502**	**0.9427**	**0.9928**	**0.9856**

that are designed to handle context, maintain state, and perform logical reasoning with greater accuracy.

1.3.2 Quantitative Analysis

A similar interesting pattern can be found in the quantitative analysis by using various LLMs on the widely used, publicly available multi-turn conversation dataset, including MultiWoz (MWOZ) and M2M.

Table 1.1 presents the experimental results of each baseline compared to the performance of our best model. It is evident that GPT-3.5 [155] achieves the highest performance across all tasks in zero-shot testing, yet it still significantly lags behind our model. In the ID and SF tasks, LLaMa's performance is notably inferior to that of Gemma and GPT, indicating that factors such as architecture, training data, and training methods impact LLM performance beyond just the number of parameters.

Within the LLaMa series, the number of parameters doesn't always correlate with better performance; the 7b model occasionally outperforms the 13b and 70b models. It's worth mentioning that only the 70b model was used with 4-bit quantization.

Across both tasks, LLMs sometimes generate out-of-scope class names despite being provided with all class names. Additionally, in the SF task, LLMs often fail to produce answers matching the length of the original text. Despite prompts indicating that no explanation is necessary for efficiency, LLMs occasionally generate explanations. These observations suggest that LLMs do not fully understand the input.

1.4 Conclusion

In this chapter, we have journeyed through the intricate landscape of Natural Language Understanding (NLU) and examined the limitations of Large Language Models (LLMs) like GPT-3.5 and LLaMa. While these advanced models have revolutionized conversational systems and demonstrated remarkable feats in zero-shot tasks, they also exhibit significant shortcomings that underscore the ongoing necessity for robust NLU techniques.

We observed that factors beyond the sheer number of parameters—such as architecture, training data, and methodologies—play crucial roles in determining model performance. Despite GPT-3.5's impressive performance, our experiments revealed that it still falls short when compared to our optimized models, particularly in tasks that require nuanced understanding and logical reasoning. These models sometimes generate out-of-scope class names, fail to match the length of responses to the original input, and misunderstand key instructions, indicating that they do not fully grasp the input context.

In the following chapters, we will delve deeper into the detailed techniques and applications pushing the boundaries of NLU. We will explore state-of-the-art methods, examine practical applications, and uncover how emerging technologies are addressing the current limitations of LLMs. Join us as we continue to navigate the fascinating and complex world of natural language understanding, striving for a future where machines can truly comprehend and interact with human language.

Chapter 2
Prerequisites and Glossary for Natural Language Understanding

Welcome to Chap. 2 of our book on Natural Language Understanding (NLU) for conversation agents. This chapter is a foundational guide, combining essential prerequisites and a comprehensive glossary of critical terminologies drawn from Chaps. 1, 3, 4, 5, and 6. These terms are fundamental to understanding the intricate workings of modern NLU systems and their application in developing intelligent dialogue systems. In our exploration of NLU, we delve into a diverse array of concepts and technologies from artificial intelligence, machine learning and Natural Language Processing. Whether you are new to the field or seeking to deepen your understanding, this chapter aims to provide clarity and insight into the foundational principles that underpin effective NLU and dialogue management. The glossary presented here defines and elaborates on terms essential for comprehending NLU and its practical applications. Each definition is accompanied by a detailed description that illuminates its significance in dialogue systems and illustrative examples showcasing real-world implementations and scenarios. By understanding these foundational concepts and terminologies, readers can navigate through the subsequent chapters, which explore advanced topics in depth. From *Multi-Head Attention* and *Task-oriented Chatbots* to *Integrated Learning* and *Dialogue State*, these terms provide a comprehensive framework for understanding and developing NLU systems.

Annotate (Annotation) Annotation in the context of natural language processing refers to the process of adding notes or labels to text data, which help machines understand the structure, meaning, and context of the text.

Detailed Description Annotation is essential for training machine learning models. It involves marking parts of the text with labels such as named entities (like names of people, organizations, or locations), parts of speech (nouns, verbs, adjectives), and other linguistic features. This labelled data serves as a foundation for teaching models to recognize and understand similar patterns in new, unlabelled text.

S. C. Han et al., *Natural Language Understanding in Conversational AI with Deep Learning*, https://doi.org/10.1007/978-3-031-74364-1_2

Example For instance, in the sentence "Alice went to Paris," an annotator might label "Alice" as a person (PER) and "Paris" as a location (LOC). This helps the machine learning model to learn these entities' roles and recognize them in future texts.

Annotated Data Annotated data refers to datasets that have been enriched with labels or notes, providing additional information about the data's content and structure, which is crucial for training machine learning models.

Detailed Description Annotated data is a key resource in supervised learning, where the model learns from input-output pairs. These annotations guide the learning process by explicitly showing the model what to look for and how to interpret the data. High-quality annotated data is vital for the accuracy and performance of natural language understanding systems.

Example Consider a dataset of customer reviews with sentiment labels. Each review might be annotated as positive, negative, or neutral. These labels help the model to learn how to classify new reviews based on their sentiment.

Attention Mechanism The attention mechanism is a technique used in neural networks, especially in sequence-based models, to dynamically focus on the most relevant parts of the input sequence when making predictions.

Detailed Description The attention mechanism allows models to weigh different parts of the input data differently, giving more importance to some parts while ignoring others. This is particularly useful in tasks like translation, where certain words in a sentence may be more critical for understanding the context than others. It helps the model to handle long-range dependencies and improve performance on various tasks.

Example In machine translation, when translating the sentence "The cat sat on the mat" from English to French, the attention mechanism helps the model to align "cat" with "chat" and "mat" with "tapis," ensuring accurate translation.

Attention Score The attention score is a value that indicates the importance of a particular part of the input data relative to others, as determined by the attention mechanism in a neural network.

Detailed Description Attention scores are calculated during the training of the model and indicate how much focus the model should place on each part of the input data when making a prediction. These scores are used to create weighted combinations of the input features, which can significantly improve the model's ability to understand and process complex data.

Example In a sentiment analysis task, if the attention mechanism assigns higher scores to words or phrases that express strong emotions or sentiments, the model can more accurately determine the overall sentiment of the text. For instance, in a product review, words like "excellent," "terrible," or "disappointed" might

receive higher attention scores, helping the model to better classify the review as positive or negative.

Benchmark In machine learning, a benchmark is a standard or point of reference against which models or algorithms can be compared to measure their performance.

Detailed Description Benchmarks are used to evaluate and compare the effectiveness of different models on specific tasks. They often involve standardized datasets and evaluation metrics, allowing researchers and developers to understand how well a model performs relative to others and to identify areas for improvement.

Example A well-known benchmark in natural language processing is the General Language Understanding Evaluation (GLUE) benchmark, which includes a variety of tasks like sentiment analysis and question answering. Models are evaluated on their performance across these tasks to provide a comprehensive assessment.

Binary Classification Binary classification is a type of classification task where the objective is to assign one of two possible classes to each input.

Detailed Description In binary classification, models are trained to differentiate between two classes, such as "spam" and "not spam" in email filtering, or "positive" and "negative" in sentiment analysis. This is one of the simplest and most common types of classification tasks in machine learning.

Example An example of binary classification is a model that predicts whether an email is spam or not. The input data (email content) is analyzed, and the model assigns a label (spam or not spam) based on its training.

Capsule Network A capsule network is a type of artificial neural network that aims to better model hierarchical relationships within data, addressing some limitations of traditional convolutional neural networks (CNNs).

Detailed Description Capsule networks use groups of neurons, called capsules, that output a vector rather than a scalar value. These vectors represent different properties of objects and their instantiation parameters, such as pose, texture, and spatial relationship. Capsule networks can better understand the spatial hierarchies and improve the robustness of models to transformations in the input data.

Example In image recognition, a capsule network might better recognize a rotated or scaled image of a cat compared to a traditional CNN, as it understands the spatial relationships between the cat's features more effectively.

Chatbot A chatbot is a software application designed to simulate human conversation, often used in customer service, information retrieval, and other interactive applications.

Detailed Description Chatbots can be rule-based or powered by artificial intelligence. Rule-based chatbots follow predefined rules and patterns to respond to user inputs, while AI-powered chatbots use natural language processing and machine learning to understand and generate more complex and natural responses. They are commonly used in Web sites, messaging apps, and customer support systems to provide immediate assistance and automate repetitive tasks.

Example When you visit an online store and see a chat window pop up asking if you need help, the software behind that chat window is likely a chatbot. It can answer questions about products, help with order tracking, and even assist with returns.

Cluster Merger Algorithm A cluster merger algorithm is a technique used in clustering analysis to combine smaller, similar clusters into larger ones, enhancing the overall structure and coherence of the clustered data.

Detailed Description Clustering is the process of grouping similar data points together. A cluster merger algorithm iteratively examines the clusters formed and merges those that are similar or close to each other based on a predefined criterion, such as distance or similarity measures. This helps in forming more meaningful and less fragmented clusters.

Example In customer segmentation, a cluster merger algorithm might combine clusters of customers with similar purchasing patterns to create broader segments that can be targeted with specific marketing strategies.

Conditional Random Field (CRF) A Conditional Random Field (CRF) is a type of statistical modelling method often used for structured prediction, where the output depends on the entire input sequence rather than just individual input components.

Detailed Description CRFs are used in natural language processing tasks such as part-of-speech tagging, named entity recognition, and information extraction. Unlike other models that make independent predictions for each part of the input, CRFs consider the context of the entire input sequence, which helps in making more accurate predictions by capturing dependencies between adjacent labels.

Example In named entity recognition, a CRF can be used to label words in a sentence as person names, organizations, locations, etc., while considering the context provided by surrounding words to improve accuracy.

Context Management Context management refers to the techniques and processes used to handle and utilize context information in conversation agents to maintain coherent and relevant interactions.

Detailed Description Effective context management enables conversation agents to remember and appropriately use information from previous interactions. This can include user preferences, past questions and answers, and other relevant data. Proper context management is crucial for providing a seamless and natural conversational experience.

Example If a user asks a virtual assistant, "What's the weather like today?" followed by "And tomorrow?", the system uses context management to understand that the second question is also about the weather, but for the next day.

Context Retention Context retention is the ability of a conversation agent to remember and utilize information from earlier parts of a conversation to provide coherent and relevant responses in subsequent interactions.

Detailed Description Maintaining context over multiple turns in a conversation is essential for creating natural interactions. This involves storing and retrieving relevant information as the conversation progresses, enabling the agent to provide responses that are contextually appropriate and informed by prior exchanges.

Example In a customer service chatbot, if a user first asks about a product's availability and then inquires about shipping options, context retention allows the chatbot to remember the specific product being discussed when providing shipping information.

Context Window A context window in natural language processing refers to the span of text surrounding a target word or phrase that is considered when analyzing or predicting the meaning or function of that word or phrase.

Detailed Description The context window is crucial for understanding the meaning of words that depend on their surroundings. By examining the words immediately before and after a target word, models can better grasp its meaning and how it relates to the overall sentence or document. The size of the context window can vary depending on the application and the complexity of the text.

Example In the sentence "She opened the door with a key," a context window might include the words "opened," "the," and "with" to help determine that "key" refers to an object used to unlock the door.

Contextual Continuity Contextual continuity refers to the smooth and logical flow of context within a conversation, ensuring that responses are coherent and relevant to the preceding dialogue.

Detailed Description Maintaining contextual continuity is essential for creating natural and effective conversations. It involves tracking and updating the context as the conversation progresses, ensuring that each response logically follows from the previous interactions. This helps in maintaining user engagement and providing accurate information.

Example If a user is asking a series of questions about a travel itinerary, contextual continuity ensures that the conversation agent can provide consistent and relevant information about flights, hotels, and activities without losing track of the overall context.

Contextual Dependency Contextual dependency refers to the reliance of a word, phrase, or sentence on its surrounding text to convey its complete meaning.

Detailed Description Many words and phrases in natural language depend on the context in which they appear to be fully understood. Contextual dependency is a critical consideration in natural language processing, as it allows models to interpret the meaning based on the surrounding text, rather than in isolation.

Example The word "bank" can refer to a financial institution or the side of a river. Its meaning depends on the surrounding words, such as in "She deposited money at the bank" versus "They had a picnic on the river bank."

Contextual Reference Contextual reference involves using information from the surrounding text or conversation to resolve ambiguities and provide specific, relevant responses.

Detailed Description Contextual reference is essential for understanding and generating natural language. It helps in identifying what or whom a particular word or phrase refers to, based on the context. This is especially important in conversation agents to ensure that responses are accurate and relevant to the ongoing dialogue.

Example In a conversation, if someone says, "I spoke to John yesterday. He said he will join us," the pronoun "he" refers to "John" based on contextual reference.

Contextualized Word Representation Contextualized word representation is a method in natural language processing where the meaning of a word is determined based on the context in which it appears.

Detailed Description Unlike traditional word embeddings that assign a single representation to each word regardless of context, contextualized word representations adjust the representation based on the surrounding text. This allows models to understand the nuances of language better and disambiguate words that have multiple meanings.

Example The word "bat" can mean a flying mammal or a piece of sports equipment. In the sentence "The bat flew out of the cave," a contextualized word representation would interpret "bat" as the animal, while in "He swung the bat and hit the ball," it would interpret "bat" as the sports equipment.

Continuous Learning Continuous learning in machine learning refers to the ability of a model to keep learning from new data and experiences after its initial training.

Detailed Description Continuous learning enables models to adapt to new information and improve over time without requiring retraining from scratch. This is particularly useful in dynamic environments where data evolves, such as personalized recommendation systems or autonomous vehicles. Continuous learning can help maintain the model's relevance and accuracy.

Example A spam detection system that continuously learns can adapt to new spam tactics and remain effective over time, improving its ability to filter out unwanted emails as new patterns emerge.

Conversation Topic Detection Conversation topic detection is the process of identifying the main subject or theme of a conversation within a dialogue system.

Detailed Description This involves analyzing the content of the conversation to determine the topics being discussed. Effective topic detection helps conversational agents provide relevant responses, maintain context, and manage conversations more effectively. It can also aid in summarizing and categorizing dialogues for further analysis.

Example In a customer service chatbot, conversation topic detection might identify that the user is inquiring about billing issues, enabling the bot to provide specific and relevant information related to billing.

Conversation-Level Domain The conversation-level domain refers to the broader context or subject matter that a conversation is centered around, guiding the direction and nature of the dialogue.

Detailed Description Understanding the conversation-level domain is crucial for maintaining relevance and coherence in interactions. It involves recognizing the overall topic or field the conversation pertains to, such as healthcare, finance, or travel. This helps in providing domain-specific information and responses that align with the user's needs and expectations.

Example In a virtual health assistant, recognizing that the conversation-level domain is medical advice helps the agent provide accurate and relevant health information, avoiding off-topic responses.

Conversational Agent A conversational agent is a software application designed to interact with users through natural language, often performing tasks, providing information, or engaging in dialogue.

Detailed Description Conversational agents can be powered by rule-based systems or advanced artificial intelligence techniques, including natural language processing and machine learning. They are commonly used in customer service, virtual assistants, and interactive platforms to simulate human conversation and perform automated tasks.

Example Popular examples of conversational agents include virtual assistants like Amazon's Alexa, Apple's Siri, and chatbots used on customer service Web sites to help users navigate services and answer queries.

Conversational AI Conversational AI refers to the technologies and methods used to enable machines to understand, process, and respond to human language in a conversational manner.

Detailed Description Conversational AI encompasses natural language processing, machine learning, and speech recognition technologies to create systems that can engage in dialogue with users. These systems can handle a variety of tasks, from simple question answering to complex multi-turn conversations, providing personalized and contextually relevant responses.

Example A customer service chatbot that can understand customer issues, ask follow-up questions, and provide solutions, all while maintaining a natural conversational flow, is an application of conversational AI.

Convolutional Neural Network (CNN) A Convolutional Neural Network (CNN) is a type of deep learning model particularly effective for processing grid-like data, such as images.

Detailed Description CNNs use convolutional layers to automatically and adaptively learn spatial hierarchies of features from input data. They are designed to recognize patterns such as edges, textures, and shapes by applying filters that slide over the input data. CNNs are widely used in image and video recognition, medical image analysis, and natural language processing tasks that can be represented as spatial data.

Example CNN can be used to classify text into different categories by learning to recognize patterns in the sequences of words. For instance, in sentiment analysis, a CNN can learn to detect specific phrases and word combinations that indicate positive or negative sentiment, enabling the model to accurately classify a given piece of text as expressing positive, negative, or neutral sentiment.

Crowd-Sourcing Crowd-sourcing is the practice of obtaining input, ideas, or services by soliciting contributions from a large group of people, typically from an online community.

Detailed Description In the context of machine learning and data annotation, crowd-sourcing involves gathering large amounts of labelled data by engaging a broad and diverse group of contributors. This approach leverages the collective intelligence and efforts of many individuals to perform tasks that would be time-consuming or difficult for a small team. Crowd-sourcing is often used to collect training data for supervised learning models.

Example Platforms like Amazon Mechanical Turk allow businesses to crowd-source data annotation tasks, such as labelling images, transcribing audio, or categorizing text, by distributing the work to a large pool of online workers.

Customer Service System A customer service system is a set of tools and processes designed to assist customers in resolving issues, obtaining information, and enhancing their overall experience with a company or service.

Detailed Description These systems often include various channels such as phone support, email, live chat, and automated chatbots. The goal is to provide

timely and efficient assistance to customers, improving satisfaction and loyalty. Advanced customer service systems leverage technologies like artificial intelligence and natural language processing to handle inquiries more effectively and personalize interactions.

Example An online retailer might use a customer service system that includes a chatbot to answer common questions, a live chat feature for more complex issues, and a ticketing system to track and resolve customer complaints.

Decision Tree A decision tree is a type of algorithm used in machine learning and statistics that models decisions and their possible consequences, including chance event outcomes and resource costs.

Detailed Description Decision trees use a tree-like model of decisions, where each internal node represents a test or attribute, each branch represents the outcome of the test, and each leaf node represents a class label or decision. They are simple to understand and interpret, making them popular for both classification and regression tasks. Decision trees can handle both numerical and categorical data.

Example In a medical diagnosis application, a decision tree might help doctors determine a diagnosis by asking a series of yes/no questions about a patient's symptoms, leading to a final diagnosis at the leaf nodes.

Deep Learning Deep learning is a subset of machine learning that involves neural networks with many layers (deep neural networks), which are capable of learning from large amounts of data.

Detailed Description Deep learning models are designed to automatically discover representations and patterns in data, often achieving superior performance in tasks like image recognition, natural language processing, and speech recognition. These models consist of multiple layers of interconnected neurons, with each layer learning increasingly abstract features from the data.

Example A deep learning model might be used in a self-driving car to recognize and respond to traffic signals, pedestrians, and other vehicles by processing camera images in real time.

Delexicalization Delexicalization is a technique used in natural language processing where specific words or phrases in a sentence are replaced with placeholders to generalize the sentence structure.

Detailed Description This process involves substituting variable information such as names, dates, and locations with generic tokens. Delexicalization is often used to simplify and standardize input data, making it easier for models to recognize patterns and learn from the data without being distracted by specific, infrequent terms.

Example The sentence "John will meet Mary in London on Monday" might be delexicalized to "X will meet Y in Z on W," where X, Y, Z, and W are placeholders.

Dependency Parsing Dependency parsing is a technique in natural language processing that involves analyzing the grammatical structure of a sentence to identify the dependencies between words.

Detailed Description Dependency parsing aims to determine how words in a sentence are related to each other, mapping out the syntactic structure in terms of a dependency tree. Each node in the tree represents a word, and the edges represent the dependencies between words. This is essential for understanding the meaning of a sentence and is used in various applications like machine translation and information extraction.

Example In the sentence "The cat sat on the mat," dependency parsing would identify "cat" as the subject of the verb "sat," and "mat" as the object of the preposition "on."

Dialog State Tracking Dialog state tracking is the process of maintaining and updating the state of a conversation in a dialogue system to manage context and guide future interactions.

Detailed Description Dialog state tracking involves keeping track of user goals, system actions, and the history of the conversation. This helps the system to understand the current context and provide relevant and coherent responses. It is crucial for creating effective conversational agents that can handle multi-turn interactions and maintain context over the course of the conversation.

Example In a restaurant reservation system, dialog state tracking would involve keeping track of information like the number of guests, reservation time, and user preferences as the conversation progresses.

Dialogue Act A dialogue act is a function or purpose that a speaker's utterance serves within a conversation, such as asking a question, making a statement, or giving a command.

Detailed Description Dialogue acts are crucial for understanding the intent behind a user's words and guiding the appropriate response in a dialogue system. Recognizing dialogue acts helps in structuring conversations and maintaining coherent interactions.

Example In the sentence "Can you tell me the time?", the dialogue act is a question, as the speaker is seeking information.

Dialogue State Dialogue state refers to the current context or status of an ongoing conversation, including the user's goals, preferences, and any relevant information that has been exchanged.

Detailed Description Maintaining an accurate dialogue state is essential for managing context and ensuring that the system responds appropriately to the user. It involves tracking various elements like the history of interactions, unresolved queries, and any specified user details.

Example In a hotel booking conversation, the dialogue state would include information such as the check-in date, number of guests, and room type preferences as provided by the user.

Domain-Specific Domain-specific refers to techniques, applications, or systems that are tailored to operate within a particular area of knowledge or industry.

Detailed Description Domain-specific systems are designed to address the unique requirements and challenges of a specific field, such as healthcare, finance, or education. These systems incorporate specialized knowledge and terminology relevant to the domain, enabling more accurate and effective solutions.

Example A domain-specific natural language processing system for healthcare might be trained to recognize medical terminology and understand patient symptoms and diagnoses.

Dynamic Dialogue Management Dynamic dialogue management refers to the ability of a conversational agent to adaptively manage the flow of conversation based on real-time context and user inputs.

Detailed Description This involves adjusting the conversation strategy as new information is obtained, ensuring that the dialogue remains coherent and relevant. Dynamic dialogue management helps in handling unexpected user responses and changing contexts, providing a more flexible and natural interaction experience.

Example If a user starts talking about a different topic in the middle of a conversation with a customer service chatbot, dynamic dialogue management allows the system to shift focus and respond appropriately to the new topic.

Edge Case An edge case refers to a situation or condition that occurs at an extreme (maximum or minimum) operating parameter, often unanticipated and not commonly encountered.

Detailed Description Edge cases are important in software and system design because they can reveal vulnerabilities and potential failures that are not evident under normal conditions. Handling edge cases ensures that the system remains robust and reliable even in rare or extreme situations.

Example In a conversational agent, an edge case might be a user input that includes extremely long or nonsensical text, which the system needs to handle gracefully without crashing or providing inappropriate responses.

End-to-End End-to-end refers to a system or process where all components are integrated and function together from start to finish, without the need for intermediate steps or external intervention.

Detailed Description In machine learning, an end-to-end model is trained to perform a complete task by learning directly from input data to output results, without requiring hand-engineered features or intermediate steps. This approach simplifies the pipeline and can lead to more efficient and accurate models.

Example An end-to-end speech recognition system takes raw audio as input and directly outputs transcribed text, learning the mapping from speech to text in a single, unified model.

Ensemble Method An ensemble method is a technique in machine learning where multiple models are combined to improve overall performance and accuracy.

Detailed Description Ensemble methods leverage the strengths of different models by combining their predictions, often leading to better results than any single model alone. Common ensemble techniques include bagging, boosting, and stacking, each of which uses different strategies to integrate multiple models. In Joint NLU tasks, ensemble methods can be particularly effective by combining various NLU models to handle multiple subtasks simultaneously, such as intent detection and slot filling.

Example In a Joint NLU task, such as a virtual assistant's understanding of user queries, an ensemble method might combine the outputs of separate models trained for intent detection and slot filling. For instance, one model might excel at identifying the user's intent (e.g., booking a flight), while another model accurately extracts relevant details (e.g., departure city, destination, date). By integrating these models' predictions, the ensemble method provides a more accurate and comprehensive understanding of the user's query.

Entity Recognition Entity recognition, also known as named entity recognition (NER), is a process in natural language processing that identifies and classifies key information (entities) in text into predefined categories such as names of persons, organizations, locations, dates, and more.

Detailed Description Entity recognition helps in extracting structured information from unstructured text, making it easier to analyze and utilize in various applications. It involves identifying entities mentioned in the text and categorizing them into specific types, which is crucial for tasks like information retrieval, question answering, and text summarization.

Example In the sentence "Alice visited Paris last summer," entity recognition would identify "Alice" as a person, "Paris" as a location, and "last summer" as a date/time expression.

Feature Extraction Feature extraction is the process of transforming raw data into a set of characteristics or features that can be used for machine learning tasks.

Detailed Description The goal of feature extraction is to reduce the complexity of data while retaining the most relevant information for analysis. In natural language processing, this might involve extracting features like word counts, n-grams, or syntactic dependencies. Effective feature extraction is essential for improving the performance of machine learning models.

Example In sentiment analysis, feature extraction might involve identifying positive and negative words in a text to determine the overall sentiment.

Few-Shot Few-shot learning refers to the ability of a machine learning model to learn and generalize from a very limited number of training examples.

Detailed Description Traditional machine learning models require large amounts of data to perform well. Few-shot learning, on the other hand, aims to mimic human learning, where a model can understand and perform tasks with only a few examples. This is particularly useful in scenarios where data is scarce or expensive to obtain.

Example In an intent recognition task, a few-shot learning chatbot model might learn to recognize a new intent (e.g., ordering food) after being provided with only a few example sentences demonstrating that intent. This allows the chatbot to quickly adapt to new user requests with minimal training data.

Fine-Tune (Fine-Tuning) Fine-tuning is the process of taking a pre-trained machine learning model and making small adjustments to it using a new, often smaller dataset that is specific to a particular task.

Detailed Description Fine-tuning leverages the knowledge already gained by the model during its initial training phase and adapts it to perform well on a new task. This approach saves time and computational resources compared to training a model from scratch and often leads to better performance because the model starts from a more informed state.

Example A language model pre-trained on a large corpus of general text can be fine-tuned on a smaller dataset of medical documents to improve its performance in medical text processing tasks.

Gated Mechanism A gated mechanism in neural networks is a type of structure that controls the flow of information through the network, allowing the model to learn which information to keep and which to discard.

Detailed Description Gated mechanisms, such as those used in Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), help manage the problem of vanishing and exploding gradients in training deep networks. They work by using gates to regulate the addition or removal of information at each step of the learning process.

Example In an LSTM network, there are input, output, and forget gates that control the cell state and help in learning long-term dependencies in sequential data.

Generative Model A generative model is a type of machine learning model that learns to generate new data instances that resemble the training data.

Detailed Description Generative models are trained to understand the underlying distribution of the data and can create new examples that fit within that distribution. They are used in tasks such as image generation, text generation, and data augmentation. These models include techniques like Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs).

Example A generative model can be used for text generation. For instance, a generative model trained on a large corpus of news articles can generate realistic news articles on new topics, capturing the style and structure of the training data. This can be particularly useful for applications such as automated content creation or conversational agents that need to generate human-like responses.

Hidden Markov Model (HMM) A Hidden Markov Model (HMM) is a statistical model that represents systems which are Markov processes with hidden states.

Detailed Description HMMs are used to model time series data where the system being modelled is assumed to be a Markov process with unobserved (hidden) states. They are particularly useful in applications like speech recognition, where the observable data (audio signals) is influenced by hidden states (spoken words).

Example In speech recognition, an HMM can model the sequence of spoken words by associating each word with a sequence of hidden states that generate the observed audio signal.

Hidden State In the context of neural networks and sequence models, a hidden state is an internal state that captures information about the input sequence up to the current point.

Detailed Description Hidden states are crucial in models like Recurrent Neural Networks (RNNs) and their variants, where the state evolves as the model processes each element of the input sequence. The hidden state helps the model to maintain context and make informed predictions based on previous inputs.

Example In a language model processing the sentence "The cat sat on the mat," the hidden state at each word position captures the context of the words seen so far, helping the model predict the next word.

Hierarchical Model A hierarchical model is a type of model that organizes data or concepts into a tree-like structure with multiple levels, where higher-level concepts encompass or depend on lower-level ones.

Detailed Description Hierarchical models are used in various fields to represent relationships and dependencies in data. In machine learning and natural language processing, they can be used to model complex relationships, such as the structure of a document, where paragraphs are composed of sentences, which in turn are composed of words.

Example In topic modelling, a hierarchical model might represent topics at different levels of granularity, such as general topics at the top level and more specific subtopics at lower levels.

Hierarchical Representation Hierarchical representation refers to the way data or concepts are organized in a multi-level structure, where each level provides a different degree of abstraction or detail.

Detailed Description This type of representation allows for efficient processing and understanding of complex data by breaking it down into manageable components. In neural networks, hierarchical representations can help in learning and recognizing patterns at various levels, from simple features to complex structures.

Example In the task of parsing a sentence, the first layers of a neural network might detect basic linguistic features like parts of speech (nouns, verbs, etc.), while deeper layers recognize more complex structures like phrases and clauses. The highest layers might understand the overall meaning of the sentence, forming a hierarchical representation of the text.

Informable Slot An informable slot is a field or category in a dialogue system that a user can provide information about, often used in task-oriented dialogue systems to collect specific details required to complete a task.

Detailed Description Informable slots are crucial in systems like booking assistants or customer service chatbots, where gathering accurate information from the user is necessary to perform a service. These slots represent the pieces of information the system needs to fulfill the user's request.

Example In a restaurant booking system, informable slots might include the date, time, number of people, and restaurant location. The user provides information for these slots to make a reservation.

Integrated Learning Integrated learning involves combining multiple learning tasks or models into a unified framework, allowing them to benefit from shared information and improve overall performance.

Detailed Description This approach leverages the strengths of different models or tasks, enabling them to share knowledge and learn from each other. Integrated learning can lead to better generalization and robustness by incorporating diverse perspectives and data sources.

Example In a language understanding system, integrated learning might combine tasks like entity recognition, intent detection, and sentiment analysis into a single model, allowing each task to benefit from the shared contextual understanding.

Intent Classification Intent classification is the process of identifying the user's goal or purpose behind a given input in a natural language processing system.

Detailed Description This involves analyzing the user's utterance to determine what action they want to perform or what information they are seeking. Intent classification is a key component in conversational agents and chatbots, enabling the system to understand and respond appropriately to user requests.

Example In a virtual assistant, if a user says "Set a reminder for 3 PM," the system would classify the intent as "set reminder" and proceed to create the reminder at the specified time.

Intent Recognition Intent recognition is the broader process of understanding and interpreting the user's intent within the context of a conversation, which may involve identifying multiple intents and related entities.

Detailed Description Intent recognition involves not only classifying the user's primary intent but also understanding the nuances and additional information provided in their input. This comprehensive understanding helps in delivering more accurate and relevant responses.

Example If a user says, "Book a flight to New York for next Monday," intent recognition involves identifying the primary intent to book a flight and extracting related details like the destination (New York) and date (next Monday).

Joint Model A joint model in machine learning simultaneously handles multiple related tasks, leveraging shared information and dependencies between them to improve overall performance.

Detailed Description Joint models are designed to solve interconnected tasks together rather than independently, which can lead to better results by exploiting the relationships between tasks. This approach is common in natural language processing, where tasks like entity recognition and intent detection are often related.

Example In a dialogue system, a joint model might simultaneously perform intent classification and slot filling, ensuring that the identified intent and extracted slots are consistent and contextually accurate.

Joint Natural Language Processing (NLU) Joint Natural Language Understanding (Joint NLU) refers to models that simultaneously perform multiple natural language understanding tasks, such as intent detection and slot filling, within a unified framework.

Detailed Description Joint NLU models aim to improve the efficiency and accuracy of understanding user inputs by handling related tasks together. This integrated approach allows the model to consider the dependencies between tasks, leading to more coherent and accurate interpretations of the input data.

Example In a virtual assistant, a joint NLU model might process the utterance "Find me a nearby Italian restaurant" by identifying the intent (finding a restaurant) and filling slots (cuisine type: Italian, location: nearby) simultaneously.

Knowledge Base A knowledge base is a repository of structured information that a system uses to answer queries, provide information, and support decision-making processes.

Detailed Description Knowledge bases contain facts, rules, and relationships about specific domains or topics, which can be accessed and utilized by various applications, including search engines, chatbots, and expert systems. They enable systems to retrieve and use relevant information efficiently.

Example A knowledge base for a customer service chatbot might include information about products, troubleshooting steps, and company policies, allowing the chatbot to provide accurate and relevant responses to customer inquiries.

Knowledge Graph A knowledge graph is a structured representation of knowledge in the form of entities, their attributes, and the relationships between them, organized in a graph format.

Detailed Description Knowledge graphs are used to store and retrieve complex information by representing data in nodes (entities) and edges (relationships). They enable systems to understand and reason about data by connecting related pieces of information in a meaningful way. Knowledge graphs are widely used in search engines, recommendation systems, and AI applications.

Example An example of a knowledge graph is Google's Knowledge Graph, which helps improve search results by connecting facts about people, places, and things to provide more relevant information.

Label (Labelling) Labelling in machine learning refers to the process of assigning meaningful tags or categories to data points, indicating what they represent or what action should be taken based on them.

Detailed Description Labelling is crucial for supervised learning, where models are trained on labelled datasets to learn the relationship between input data and the corresponding output labels. This process involves annotating data with information such as class names, categories, or relevant properties.

Example In a sentiment analysis task, labelling might involve tagging text reviews as "positive," "negative," or "neutral" based on the expressed sentiment.

Labelled Data Labelled data refers to datasets that have been annotated with labels, providing ground truth information that is used to train and evaluate machine learning models.

Detailed Description Labelled data is essential for supervised learning, where the model learns to predict labels based on input features. The quality and quantity of labelled data significantly impact the performance and accuracy of machine learning models.

Example A dataset of images labelled with categories such as "cat," "dog," and "bird" is an example of labelled data used to train an image classification model.

Large Language Model (LLM) A large language model (LLM) is a type of neural network-based model that has been trained on vast amounts of text data to understand and generate human language.

Detailed Description LLMs, such as GPT-3, are capable of performing various natural language processing tasks, including text generation, translation, summarization, and question answering. These models leverage their extensive training data to produce coherent and contextually relevant responses, making them useful in a wide range of applications.

Example GPT-3, which can generate human-like text based on prompts, is a prominent example of a large language model.

Logistic Regression Logistic regression is a statistical method used for binary classification that models the probability of a binary outcome based on one or more predictor variables.

Detailed Description Logistic regression uses the logistic function to transform the linear combination of input features into a probability score between 0 and 1. This score indicates the likelihood of the outcome belonging to the positive class. It is widely used for tasks such as spam detection, disease diagnosis, and credit scoring.

Example In a spam detection system, logistic regression might be used to predict whether an email is spam (1) or not spam (0) based on features like word frequency and sender information.

Long-Range Conversation Long-range conversation refers to dialogue systems' ability to maintain context and coherence over extended interactions, involving multiple exchanges between the user and the system.

Detailed Description Long-range conversation capabilities are crucial for creating natural and effective dialogue systems that can remember previous interactions and respond appropriately in the context of an ongoing conversation. This involves tracking the conversation history and managing the dialogue state over long interactions.

Example A customer support chatbot that can remember a user's previous inquiries about an order and provide follow-up assistance in subsequent interactions demonstrates long-range conversation capabilities.

Long-Range Dependency Long-range dependency refers to the relationships or dependencies between elements in a sequence that are far apart from each other in the input data.

Detailed Description In natural language processing, capturing long-range dependencies is important for understanding the context and meaning of text, especially in lengthy documents or conversations. Models that effectively handle long-range dependencies can better understand and generate coherent text.

Example In the sentence "The book that John, who lives in New York, gave me is fascinating," the model needs to understand the long-range dependency between "book" and "is fascinating" to grasp the correct meaning.

Machine Learning (ML) Machine learning (ML) is a field of artificial intelligence that involves developing algorithms and statistical models that enable computers to learn from and make predictions or decisions based on data.

Detailed Description Machine learning encompasses a variety of techniques, including supervised learning, unsupervised learning, and reinforcement learning. These methods allow systems to improve their performance on a given task over time without being explicitly programmed for each specific task.

Example Examples of machine learning applications include image recognition, speech recognition, recommendation systems, and autonomous driving.

Machine Translation Machine translation is a subfield of computational linguistics that focuses on the use of software to translate text or speech from one language to another.

Detailed Description Machine translation systems can be based on rule-based, statistical, or neural methods. Neural machine translation (NMT) models, which use deep learning techniques, have significantly improved translation quality by learning to capture the context and nuances of the source and target languages.

Example A neural machine translation model might be used to translate a medical report from Spanish to English. For example, the original Spanish text "El paciente presenta síntomas de fiebre y tos persistente" would be accurately translated to "The patient presents symptoms of fever and persistent cough," ensuring that medical professionals can understand the patient's condition without any loss of critical information.

Memory Network A memory network is a type of neural network architecture that incorporates a memory component to store and retrieve information, enabling the model to use long-term information to enhance its performance on tasks.

Detailed Description Memory networks are designed to address tasks that require reasoning over long-term dependencies and large contexts. They consist of a memory component that can store information and a mechanism for reading from and writing to this memory, allowing the network to utilize past information effectively.

Example In a question-answering system, a memory network might store previous dialogue contexts or external knowledge sources, allowing it to answer questions more accurately by referring to relevant stored information.

Multi-Head Attention Multi-head attention is a mechanism used in neural networks, particularly in transformer models, which allows the model to focus on different parts of the input sequence simultaneously.

Detailed Description Multi-head attention works by applying multiple attention mechanisms (heads) in parallel, each focusing on different parts of the input sequence. The results are then concatenated and transformed to produce the final output. This approach helps the model capture various aspects of the data and improves performance on tasks like translation and text generation.

Example In a transformer-based virtual assistant for intent classification, multi-head attention enables the model to consider multiple words and phrases in a user query. For instance, in the query "Book a flight to New York next Friday," different heads might focus on "Book a flight," "New York," and "next Friday," allowing the model to accurately identify the user's intent and relevant details for booking the flight.

Multi-turn Context Management Multi-turn context management refers to the capability of a dialogue system to maintain and utilize context over multiple exchanges in a conversation, ensuring coherence and relevance throughout.

Detailed Description Effective multi-turn context management allows a conversational agent to remember previous interactions and use this information to inform future responses. This is crucial for maintaining natural and engaging conversations, especially in complex tasks that require multiple steps or follow-up questions.

Example In a customer service chatbot, multi-turn context management ensures that if a user starts a conversation about a product issue, the chatbot remembers the product details and previous troubleshooting steps throughout the interaction.

Multi-turn Conversation A multi-turn conversation is an interaction between a user and a system that involves multiple exchanges or turns, where the context and state of the conversation are carried over across turns.

Detailed Description Multi-turn conversations require the system to handle context and maintain coherence across multiple user inputs and system responses. This is important for tasks that cannot be completed in a single interaction, such as booking a flight or troubleshooting an issue.

Example When a user interacts with a travel booking assistant, the conversation might include several turns to gather information about travel dates, destinations, preferences, and payment details.

Multi-turn Dialogue Multi-turn dialogue refers to an extended interaction between a user and a system, involving several exchanges to achieve a specific goal or complete a task.

Detailed Description Multi-turn dialogues require the system to understand and manage the context over multiple turns, ensuring that each response is relevant and builds upon previous exchanges. This is essential for complex tasks that involve multiple steps and require detailed information gathering and processing.

Example In a technical support chatbot, a multi-turn dialogue might involve diagnosing a problem, suggesting solutions, and providing step-by-step instructions over several turns.

Multi-turn Dialogue Management Multi-turn dialogue management involves techniques and processes used to handle and maintain coherent interactions across multiple turns in a conversation.

Detailed Description This includes tracking the dialogue state, managing context, handling user intents, and generating appropriate responses. Effective multi-turn dialogue management ensures that the system can maintain a natural flow of conversation, handle interruptions, and adapt to new information provided by the user.

Example In a restaurant reservation system, multi-turn dialogue management would involve keeping track of user preferences, available time slots, and confirming the reservation details over multiple turns.

Multi-turn Intent Understanding Multi-turn intent understanding refers to the ability of a system to accurately identify and interpret user intents over multiple exchanges in a conversation.

Detailed Description This involves recognizing user goals and intents across multiple turns, even if they are expressed indirectly or spread out over several exchanges. It is crucial for providing relevant and coherent responses in complex interactions that span multiple turns.

Example In a healthcare chatbot, multi-turn intent understanding might involve recognizing that a user asking about symptoms, medication, and appointment scheduling is seeking comprehensive medical advice.

Multi-turn NLU Multi-turn Natural Language Understanding (NLU) involves comprehending and processing user inputs over multiple turns in a conversation, ensuring that the system maintains context and responds appropriately.

Detailed Description Multi-turn NLU systems must handle the continuity of dialogue, track user intents and entities across turns, and update the dialogue state accordingly. This capability is essential for creating natural and effective conversational agents that can handle complex interactions.

Example In a virtual assistant, multi-turn NLU ensures that if a user starts a conversation about booking a flight and later asks about hotel options, the system remembers the travel details provided earlier.

Multimodal Learning Multimodal learning involves integrating and processing information from multiple modalities, such as text, images, audio, and video, to improve learning and understanding.

Detailed Description Multimodal learning enables models to leverage different types of data and their interactions, leading to more robust and accurate predictions. This approach is useful in applications where information is conveyed through various channels, and understanding requires combining these different sources.

Example In a multimedia search engine, multimodal learning allows the system to process and understand both text queries and images, providing more relevant search results by combining information from both modalities.

N-gram An N-gram is a contiguous sequence of N items from a given sample of text or speech, commonly used in natural language processing to analyse and model language.

Detailed Description N-grams can consist of words, characters, or phonemes. They are used to capture the context and dependencies in the data. The choice of N determines the length of the sequence: bigrams (N=2), trigrams (N=3), etc. N-grams are fundamental in tasks like language modelling, text generation, and speech recognition.

Example In the sentence "The quick brown fox," the bigrams are "The quick," "quick brown," and "brown fox."

Named Entity Recognition (NER) Named Entity Recognition (NER) is a subtask of information extraction that involves identifying and classifying named entities in text into predefined categories such as person names, organizations, locations, dates, and more.

Detailed Description NER helps in structuring unstructured text by extracting and categorizing key information. This process is essential for various applications, including information retrieval, question answering, and summarization. NER systems use linguistic features and machine learning models to accurately identify entities.

Example In the sentence "Barack Obama was born in Hawaii," NER would identify "Barack Obama" as a person and "Hawaii" as a location.

Natural Language Processing (NLP) Natural Language Processing (NLP) is a field of artificial intelligence that focuses on the interaction between computers and humans through natural language, enabling machines to understand, interpret, and generate human language.

Detailed Description NLP encompasses a range of tasks, including text analysis, sentiment analysis, machine translation, speech recognition, and dialogue systems. By leveraging techniques from linguistics, computer science, and machine learning, NLP aims to bridge the gap between human communication and computer understanding.

Example Applications of NLP include virtual assistants like Siri and Alexa, which can understand and respond to spoken commands, and translation services like Google Translate, which convert text from one language to another.

Natural Language Understanding (NLU) Natural Language Understanding (NLU) is a branch of artificial intelligence that focuses on enabling computers to understand, interpret, and respond to human language in a way that is both meaningful and useful.

Detailed Description NLU is a critical component of conversation agents, like chatbots and virtual assistants. It allows these systems to process input from users, understand the context and intent behind the words, and generate appropriate responses. Unlike simple keyword matching, NLU involves understanding the structure of sentences, identifying entities (like names, dates, and places), and grasping the nuances of human language, such as sarcasm or idioms.

Example When you ask a virtual assistant, "What's the weather like in London tomorrow?" the NLU component will:

- Recognize "weather" as the subject of interest
- Identify "London" as the location
- Understand "tomorrow" as the time frame
- Use this understanding to provide a relevant weather forecast

Neural Network (NN) A Neural Network (NN) is a computational model inspired by the way biological neural networks in the human brain process information, consisting of interconnected nodes (neurons) that work together to solve specific problems.

Detailed Description Neural networks are composed of layers: an input layer, one or more hidden layers, and an output layer. Each neuron receives inputs, processes them using weighted connections, and passes the output to the next layer. Neural networks are capable of learning from data through a process called training, where the weights are adjusted to minimize error.

Example A neural network might take text data as input, process it through multiple layers to detect linguistic features like words and phrases, understand context and sentiment, and finally output the classification of the sentiment (e.g.,

positive, negative, or neutral). For instance, given a product review "This phone is amazing and has great battery life," the neural network would analyze the text and classify the sentiment as positive

Numerical Representation Numerical representation refers to the conversion of text or other data types into numerical format that can be processed by machine learning models.

Detailed Description In natural language processing, numerical representation involves encoding words, phrases, or entire documents into vectors of numbers. Common methods include word embeddings like Word2Vec, GloVe, and more advanced representations like BERT. These representations capture semantic meaning and relationships between words.

Example The word "king" might be represented as a vector of numbers that captures its relationship with other words like "queen," "man," and "royalty."

Out-Of-Vocabulary (OOV) Issue The Out-Of-Vocabulary (OOV) issue occurs when a model encounters words or terms during inference that were not present in the training data.

Detailed Description OOV words can pose a challenge for natural language processing models, as these models rely on having seen words during training to understand their meaning and context. Techniques to handle OOV words include using subword tokenization, character-level models, or leveraging contextual embeddings from models like BERT.

Example If a model trained on news articles encounters the term "cryptozoology" for the first time, it might struggle to process it due to the OOV issue.

Open-Domain Conversation Open-domain conversation refers to dialogue systems designed to engage in general conversations on a wide range of topics without being limited to a specific domain or task.

Detailed Description Open-domain conversations are characterized by their flexibility and breadth, allowing participants to discuss anything from current events and personal hobbies to abstract concepts and general knowledge. These conversations require a deep understanding of language and context, as well as the ability to dynamically adapt to new topics introduced by the participants. Unlike closed-domain conversations, which are limited to a particular area of knowledge or a specific task, open-domain conversations do not have predefined boundaries, enabling a more natural and fluid exchange of information.

Example Imagine two people engaging in an open-domain conversation. They might start by talking about the weather, with one person saying, "It's been unusually hot this summer." The other might respond, "Yes, it reminds me of the heatwave we had a few years ago. By the way, have you read any good books lately?" This shifts the topic to literature, where they can discuss their favorite

authors and recent reads. As the conversation flows, they might even segue into discussing advancements in renewable energy or sharing travel experiences. The key aspect of this open-domain conversation is the seamless transition between diverse topics, showcasing the participants' ability to engage on a wide array of subjects without any specific focus.

Optimal Boundary (Hyperplane) In machine learning, an optimal boundary, or hyperplane, refers to the decision surface that best separates different classes in the feature space, often used in classification tasks.

Detailed Description The optimal boundary is determined by algorithms like Support Vector Machines (SVMs), which find the hyperplane that maximizes the margin between different classes. This boundary helps in classifying new data points by determining on which side of the hyperplane they fall.

Example In a binary classification problem with two features, the optimal boundary might be a line that separates data points into two categories with maximum margin between the closest points of each class.

Part-of-Speech Tag A part-of-speech (POS) tag is a label assigned to a word in a sentence that indicates its grammatical role, such as noun, verb, adjective, etc.

Detailed Description POS tagging is a fundamental task in natural language processing, helping models understand the syntactic structure of sentences. POS tags are used in various applications, including parsing, named entity recognition, and text-to-speech systems.

Example In the sentence "The quick brown fox jumps over the lazy dog," the word "quick" is tagged as an adjective (ADJ) and "jumps" as a verb (VERB).

Persona Consistency Persona consistency refers to the ability of a conversational agent to maintain a coherent and stable personality throughout an interaction or across multiple interactions.

Detailed Description Ensuring persona consistency involves the agent adhering to a set of predefined characteristics, behaviors, and responses that align with its intended personality. This is crucial for creating engaging and trustworthy interactions, particularly in long-term or frequent user interactions.

Example If a virtual assistant is designed to be friendly and informal, it should consistently use casual language and friendly expressions in all its responses.

Persona Sentence A persona sentence is a statement that defines a specific characteristic, preference, or background detail of a conversational agent, helping to shape its personality and guide its responses.

Detailed Description Persona sentences are used to give conversational agents a distinct personality by providing context about their character. These sentences

help the agent maintain consistent behavior and make interactions more engaging and relatable.

Example Examples of persona sentences might include "I love reading science fiction novels," or "I am an expert in digital marketing," which help the agent respond in ways that reflect these traits.

Pointer Network A pointer network is a type of neural network architecture that uses attention mechanisms to select elements from the input sequence as the output, rather than generating output from a fixed vocabulary.

Detailed Description Pointer networks are particularly useful in tasks where the output is a subset or reordering of the input, such as in machine translation, text summarization, and combinatorial optimization problems. The attention mechanism allows the model to "point" to relevant parts of the input when generating each part of the output.

Example In a coreference resolution task, a pointer network can be used to identify and link pronouns to their corresponding entities in a text. For example, in the sentence "Alice went to the park, and she enjoyed the sunny weather," the pointer network can point to "Alice" as the antecedent of the pronoun "she," ensuring accurate understanding and processing of the text.

Positional Encoding Positional encoding is a technique used in transformer models to inject information about the relative or absolute position of tokens in a sequence, enabling the model to capture the order of words.

Detailed Description Unlike recurrent neural networks, transformers do not inherently process sequences in order. Positional encoding provides a way to introduce sequence order information, usually by adding sine and cosine functions of different frequencies to the token embeddings. This helps the model understand the position of each word in the context of the entire sequence.

Example In a sentence like "The cat sat on the mat," positional encoding helps the transformer model know the order of the words, ensuring that "The" is recognized as the first word and "mat" as the last.

Pre-processing Pre-processing refers to the steps taken to clean and prepare raw data before it is used in a machine learning model, ensuring the data is in a suitable format for analysis.

Detailed Description Pre-processing techniques vary depending on the type of data and the specific requirements of the model. Common steps include tokenization, normalization, removing stop words, handling missing values, and feature scaling. Proper pre-processing is crucial for improving model accuracy and efficiency.

Example In text pre-processing, tokenization involves splitting a sentence into individual words or tokens, while normalization might include converting all text to lowercase and removing punctuation.

Pre-trained Model A pre-trained model is a machine learning model that has been previously trained on a large dataset and can be fine-tuned for specific tasks with smaller datasets.

Detailed Description Pre-trained models save time and computational resources by leveraging the knowledge they have already gained during the initial training phase. They are especially useful in natural language processing and computer vision, where large amounts of data and computational power are required for training from scratch.

Example BERT and GPT are examples of pre-trained models in natural language processing that can be fine-tuned for tasks like text classification, translation, and question answering.

Pre-training Pre-training is the process of training a machine learning model on a large, general-purpose dataset before fine-tuning it for specific tasks.

Detailed Description During pre-training, the model learns general features and patterns from the data, which can then be adapted to more specific tasks with additional, smaller datasets. This approach enhances the model's performance and efficiency on the target task.

Example A language model might be pre-trained on a large corpus of text from the Internet to learn general language patterns, and then fine-tuned on a specific dataset for sentiment analysis.

Pre-trained Language Model (PLM) A pre-trained language model (PLM) is a language model that has been pre-trained on a large corpus of text data and can be fine-tuned for specific natural language processing tasks.

Detailed Description PLMs leverage extensive pre-training to understand language patterns, syntax, and semantics. They can be adapted to various downstream tasks, such as text classification, machine translation, and named entity recognition, with minimal additional training.

Example GPT-3 is a one of the PLMs that can be fine-tuned for tasks like text generation, summarization, and conversational AI.

Prompt A prompt is a piece of text or input given to a model, particularly in the context of natural language processing, to elicit a specific response or to guide the generation of text.

Detailed Description Prompts are used to instruct models on what kind of output is desired. They can be questions, incomplete sentences, or specific instructions that help the model generate relevant and contextually appropriate responses.

Example In a named entity recognition (NER) task, a prompt can guide the model to identify and classify entities within a sentence. For instance, given the input text "Barack Obama was born in Hawaii," a prompt could be, "Identify and classify the named entities in the sentence." The model would then recognize and categorize "Barack Obama" as a person and "Hawaii" as a location.

Random Forest A random forest is an ensemble learning method that uses multiple decision trees to improve classification or regression accuracy.

Detailed Description In a random forest, each decision tree is trained on a random subset of the training data and features. The final output is determined by aggregating the results of all individual trees, which reduces overfitting and improves generalization.

Example Random forests are used in applications such as fraud detection, where combining the predictions of many decision trees can yield more accurate and robust results.

Recurrent Neural Network (RNN) A Recurrent Neural Network (RNN) is a type of neural network designed to process sequential data by maintaining a hidden state that captures information from previous steps in the sequence.

Detailed Description RNNs are particularly effective for tasks involving time series data, natural language processing, and speech recognition. They can model dependencies and patterns in sequences by passing information through the hidden state from one step to the next.

Example RNNs are used in language modeling, where they predict the next word in a sentence based on the previous words.

Recursive Neural Network (RecNN) A Recursive Neural Network (RecNN) is a type of neural network that applies the same set of weights recursively over a structured input, such as a tree, to capture hierarchical relationships.

Detailed Description RecNNs are particularly suited for processing data with hierarchical structures, such as sentences parsed into syntax trees. They can capture complex relationships and dependencies by recursively combining representations from subcomponents to form a representation of the whole.

Example RecNNs are used in natural language processing to parse sentences into hierarchical structures, which can then be used for tasks like sentiment analysis.

Reinforcement Learning Reinforcement learning is a type of machine learning where an agent learns to make decisions by taking actions in an environment to maximize cumulative reward.

Detailed Description The agent interacts with the environment, receives feedback in the form of rewards or penalties, and updates its strategy to improve

performance over time. Reinforcement learning is used in various applications, including robotics, game playing, and recommendation systems.

Example In a task like dialogue management for a conversational agent, reinforcement learning can be used to optimize the agent's responses. For instance, a chatbot might learn to improve its interaction quality by receiving rewards for successful user engagements and penalties for poor interactions. Over time, the agent adjusts its strategy to maximize user satisfaction and efficiency in the conversation, leading to more natural and effective dialogue management.

Requestable Slot A requestable slot in a dialogue system refers to a piece of information that the system can ask the user for in order to complete a task or provide a service.

Detailed Description Requestable slots are essential for task-oriented dialogue systems, where the system needs to gather specific information from the user, such as booking details, preferences, or personal data, to fulfill the user's request accurately.

Example In a restaurant reservation system, requestable slots might include the number of guests, reservation time, and special dietary requirements.

Richer Annotation Richer annotation refers to the practice of providing detailed and comprehensive labels or notes to data, enhancing the quality and utility of the annotated dataset.

Detailed Description Richer annotations include more detailed information about the data, such as hierarchical labels, contextual notes, or additional metadata. This improved granularity and context help machine learning models learn more effectively and perform better on complex tasks.

Example In named entity recognition, richer annotation might involve not only tagging names of people, places, and organizations but also providing contextual information like roles or relationships between entities.

Self-attention Self-attention, also known as intra-attention, is a mechanism in neural networks that allows the model to weigh the importance of different words in a sequence relative to each other when processing each word.

Detailed Description Self-attention helps models capture dependencies between words regardless of their distance in the sequence. It computes a weighted sum of all input representations for each word, enabling the model to focus on relevant parts of the input when generating the output.

Example In a sentence like "The cat that chased the mouse was hungry," self-attention can help the model understand that "cat" is the subject of "was hungry" by attending to the entire sentence context.

Semantic Representation Semantic representation is the process of converting text or other forms of data into a format that captures its meaning, enabling machines to understand and work with it effectively.

Detailed Description Semantic representations can take various forms, such as vectors in a high-dimensional space, where similar meanings are located closer together. These representations are crucial for tasks like information retrieval, question answering, and text summarization.

Example In an intent recognition task, semantic representations can help a model understand the meaning behind user queries. For example, given user inputs like "I want to book a flight" and "Can you help me find a flight?", a model using semantic representations can recognize that both queries share the same intent (booking a flight) despite the different wording. By converting these queries into similar vectors in semantic space, the model effectively captures their underlying meaning and responds appropriately.

Sentence-Level Intent Sentence-level intent refers to the main goal or purpose behind a user's sentence in a conversation, often identified in natural language processing tasks.

Detailed Description Identifying the sentence-level intent involves understanding what the user wants to achieve with their input, such as asking a question, making a request, or providing information. This is critical for guiding appropriate responses in dialogue systems.

Example In the sentence "Can you book a table for two at an Italian restaurant?", the intent is to make a reservation at an Italian restaurant.

Sentiment Analysis Sentiment analysis is the process of determining the emotional tone or attitude expressed in a piece of text, such as positive, negative, or neutral.

Detailed Description Sentiment analysis uses natural language processing techniques to identify subjective information in text. It is widely used in applications like social media monitoring, customer feedback analysis, and market research to gauge public opinion and sentiment.

Example In analyzing product reviews, sentiment analysis can determine whether the overall sentiment expressed is positive (e.g., "I love this product!") or negative (e.g., "This product is terrible.").

Shared Representation Shared representation refers to a single, unified format used to represent multiple types of data, enabling different tasks or models to utilize the same underlying information.

Detailed Description Shared representations help improve efficiency and performance in multi-task learning and transfer learning by allowing models to leverage common features and patterns. This approach can lead to better generalization across different but related tasks.

Example In a model trained for both translation and summarization, a shared representation of sentences can help the model perform both tasks more effectively by using common linguistic features.

Siamese Network A Siamese network is a type of neural network architecture that consists of two or more identical subnetworks with shared weights, typically used for tasks involving similarity or comparison.

Detailed Description Siamese networks are designed to process two different inputs and compute a similarity score between them. They are commonly used in applications like face verification, where the goal is to determine whether two images represent the same person.

Example In text similarity tasks, a Siamese network might be used to compare two sentences and determine their semantic similarity.

Single-Turn NLU Single-turn natural language understanding (Single-Turn NLU) refers to the ability of a system to understand and process user input within a single interaction, without considering previous interactions.

Detailed Description Single-Turn NLU focuses on accurately interpreting user input in isolation, making it suitable for straightforward queries and tasks that do not require context from prior interactions. It involves tasks like intent recognition and slot filling based on a single input.

Example When a user asks "What is the weather today?", Single-Turn NLU processes this query independently to provide a relevant response without needing previous context.

Slot Filling Slot filling is a process in natural language understanding where the system identifies and extracts specific pieces of information (slots) from user input that are necessary to complete a task or query.

Detailed Description Slot filling involves recognizing entities and details within a user's input that correspond to predefined slots in a dialogue system. These slots represent the key pieces of information required for the system to execute a function or respond accurately.

Example In a flight booking system, slot filling might involve extracting the departure city, destination city, travel date, and number of passengers from the user's request.

Slot Tagging Slot tagging is the process of assigning labels to words or phrases in a sentence to identify them as belonging to specific slots or categories.

Detailed Description Slot tagging is a crucial step in slot filling, where each token in the user's input is tagged with a slot label that indicates its role. This helps in structuring the input data and extracting relevant information for further processing.

Example In the sentence "Book a flight from New York to London," slot tagging would label "New York" as the departure city and "London" as the destination city.

Slot Value A slot value refers to the specific piece of information extracted from user input that fills a slot in a dialogue system, representing a key element needed to complete a task.

Detailed Description Slot values are the actual data points identified through slot filling and tagging processes. They are used by the system to execute functions, provide responses, or complete transactions as requested by the user.

Example In a restaurant reservation system, if the user says "I need a table for two at 7 PM," the slot values extracted might be "two" for the number of guests and "7 PM" for the reservation time.

State-of-the-Art State-of-the-art refers to the highest level of development, quality, or achievement in a particular field at a given time.

Detailed Description In technology and research, state-of-the-art signifies the most advanced, innovative, and effective methods or techniques currently available. It represents the cutting-edge advancements that set the benchmark for performance and capability.

Example GPT-4 is considered state-of-the-art in Natural Language Understanding (NLU) for its remarkable ability to comprehend and generate contextually accurate responses. For instance, in a complex question-answering task, GPT-4 can understand nuanced queries and provide precise answers by leveraging its extensive training on diverse datasets. Given a question like "What are the implications of climate change on global agriculture?", GPT-4 can analyze the context, extract relevant information, and formulate a comprehensive and informative response, demonstrating its advanced NLU capabilities.

Support Vector Machine (SVM) A Support Vector Machine (SVM) is a supervised learning algorithm used for classification and regression tasks, which finds the optimal hyperplane that best separates different classes in the feature space.

Detailed Description SVMs aim to maximize the margin between data points of different classes by identifying the hyperplane that has the largest distance to the nearest training data points of any class. This approach helps in achieving better generalization and accuracy in classification tasks.

Example SVMs can be used for image classification, where the algorithm separates images of cats from images of dogs by finding the optimal boundary between the two classes in the feature space.

Syntactic Pattern A syntactic pattern refers to the arrangement of words and phrases in a sentence that follows the grammatical rules of a language, helping to convey meaning.

Detailed Description Syntactic patterns involve the structure and order of components within a sentence, such as subjects, verbs, and objects. Recognizing these patterns is crucial for understanding the grammatical relationships between words and for tasks like parsing and language generation.

Example In the sentence "The quick brown fox jumps over the lazy dog," the syntactic pattern includes a subject ("The quick brown fox"), a verb ("jumps"), and an object ("over the lazy dog").

Task-Oriented Chatbot A task-oriented chatbot is a type of conversational agent designed to assist users in completing specific tasks, such as booking flights, ordering food, or scheduling appointments.

Detailed Description Task-oriented chatbots focus on understanding user intents, extracting relevant information, and performing predefined actions to fulfill user requests. They are typically integrated with back-end systems to process transactions and provide accurate, contextually relevant responses.

Example An example of a task-oriented chatbot is a virtual assistant that helps users book movie tickets by asking for the movie name, showtime, and number of seats, and then completing the booking process.

Task-Oriented Interaction Task-oriented interaction refers to a type of communication between a user and a system where the primary goal is to complete a specific task or achieve a particular outcome.

Detailed Description In task-oriented interactions, the system is designed to guide the user through a series of steps or gather necessary information to accomplish a specific objective. These interactions are typically structured and focused, aiming for efficiency and clarity.

Example When using a customer service chatbot to troubleshoot a technical issue, the interaction is task-oriented, with the chatbot asking specific questions to diagnose and resolve the problem.

Text Categorization Text categorization, also known as text classification, is the process of assigning predefined categories or labels to text documents based on their content.

Detailed Description This task involves analyzing the text and using machine learning models or rule-based approaches to classify documents into categories such as news topics, spam vs. non-spam emails, or sentiment polarity. Text categorization is widely used in information retrieval, content management, and sentiment analysis.

Example In an email filtering system, text categorization can be used to automatically classify incoming emails as "spam" or "important."

Topic Classification Topic classification is a subtask of text categorization that involves identifying and categorizing the main topics or themes present in a text document.

Detailed Description Topic classification aims to assign one or more topics to a document based on its content. This helps in organizing large collections of documents, improving search and retrieval, and enabling more targeted information dissemination.

Example A news article might be classified under topics such as "politics," "sports," or "technology" based on its content.

Transfer Learning Transfer learning is a machine learning technique where a model developed for one task is reused as the starting point for a model on a second, related task.

Detailed Description Transfer learning leverages the knowledge gained from training on a large, general-purpose dataset to improve performance on a specific, often smaller dataset. This approach can significantly reduce training time and enhance model accuracy by building on existing learned representations.

Example A neural network pre-trained on a large image dataset like ImageNet can be fine-tuned for a specific task, such as medical image classification, by using a smaller, domain-specific dataset.

Transformer A transformer model is a neural network architecture designed to process sequential data, such as natural language text, using self-attention mechanisms to capture dependencies between different parts of the input sequence.

Detailed Description Unlike traditional recurrent neural networks (RNNs), transformers do not rely on sequential processing, allowing them to process all elements of the input simultaneously. This parallelism significantly improves efficiency and performance, especially on large datasets. The transformer architecture consists of an encoder-decoder structure, where both parts are built from layers of self-attention and feedforward neural networks.

Example Transformer models like BERT and GPT-3 have achieved state-of-the-art results in various NLP tasks, including text generation, translation, and question answering.

Unified Architecture Unified architecture in machine learning refers to a single, coherent framework that integrates multiple components or models to handle various tasks or data types seamlessly.

Detailed Description A unified architecture aims to simplify the development and deployment of machine learning systems by providing a consistent and modular

structure. This approach allows for the sharing of information and resources across different tasks, leading to improved performance and efficiency.

Example A unified architecture might combine models for intent recognition, named entity recognition (NER), and sentiment analysis into a single framework. For instance, in a customer service chatbot, this unified architecture can simultaneously identify the user's intent (e.g., requesting a refund), recognize key entities (e.g., order number, product name), and determine the sentiment (e.g., frustration or satisfaction). By integrating these NLU tasks, the chatbot can provide more accurate and contextually relevant responses, enhancing the overall user experience.

Virtual Assistant A virtual assistant is an artificial intelligence-powered software application designed to assist users with tasks by understanding and responding to natural language inputs.

Detailed Description Virtual assistants use natural language processing (NLP) and machine learning to interpret user commands and provide relevant responses or actions. They can perform a wide range of functions, such as setting reminders, answering questions, controlling smart home devices, and more.

Example Popular virtual assistants include Apple's Siri, Amazon's Alexa, Google Assistant, and Microsoft's Cortana, all of which can understand voice commands and assist with various tasks.

Vocabulary In the context of natural language processing, vocabulary refers to the set of all unique words or tokens that a model is trained to recognize and understand.

Detailed Description The vocabulary of a model determines the range of words it can process and affects its ability to understand and generate text. Building an effective vocabulary involves selecting words that are representative of the language or domain of interest while balancing the size to manage computational complexity.

Example A vocabulary for a language model trained on English text might include common words, punctuation, and special tokens like "$\langle UNK \rangle$" for unknown words and "$\langle PAD \rangle$" for padding.

Word Embedding Word embedding is a technique in natural language processing where words are represented as dense vectors in a high-dimensional space, capturing their meanings and relationships based on their usage in a large corpus of text.

Detailed Description Word embeddings allow models to understand and work with text data by encoding words in a way that similar words have similar vector representations. This helps in capturing semantic relationships and improving the performance of various NLP tasks. Popular word embedding methods include Word2Vec, GloVe, and embeddings from models like BERT.

Example The words "king" and "queen" might have embeddings that are close to each other in the vector space, reflecting their semantic similarity.

Word Frequency Word frequency refers to the number of times a word appears in a given text or corpus, used as a measure of its importance or commonality.

Detailed Description Analyzing word frequency helps in understanding the distribution of words in a text, identifying common terms, and selecting features for text processing tasks. High-frequency words often include common stop words, while lower-frequency words may provide more specific and relevant information.

Example In a corpus of news articles, words like "the," "is," and "and" will have high frequencies, while words like "pandemic" or "election" may vary in frequency based on the topic.

Word-Level Slot A word-level slot in natural language understanding refers to a specific word or sequence of words in a user's input that is identified and extracted as part of slot filling to complete a task.

Detailed Description Word-level slots represent key pieces of information needed for the system to fulfill a user's request. Identifying and tagging these slots accurately is essential for the system to process and respond correctly.

Example In the sentence "Book a flight from New York to London," the words "New York" and "London" are word-level slots for departure and destination cities, respectively.

Zero-Shot Zero-shot learning is a machine learning technique where a model is trained to recognize and classify data it has never seen before, based on the knowledge it has learned from other tasks or domains.

Detailed Description Zero-shot learning enables models to generalize from known classes to new, unseen classes by leveraging semantic information or auxiliary data. This approach is valuable in scenarios where obtaining labelled data for every possible class is impractical.

Example In an intent recognition task, a zero-shot model might be trained on a variety of intents such as booking flights, checking weather, and setting reminders. If introduced to a new intent like "finding nearby restaurants," the model can accurately identify this new intent by leveraging its understanding of similar intents and the semantic relationships between them. This allows the model to handle new and unseen user queries without requiring additional training data for every possible intent.

Chapter 3
Single-Turn Natural Language Understanding

3.1 Overview

Natural Language Understanding (NLU) stands at the heart of contemporary dialogue systems, enabling machines to grasp and respond to human language with finesse. This chapter delves into the realm of single-turn NLU, a domain where the focus is on interpreting and processing user inputs within a solitary interaction. Single-turn NLU is pivotal for a myriad of applications, including virtual assistants, chatbots, and automated customer service systems, where prompt and precise responses are indispensable.

This chapter is meticulously structured to explore the essential tasks in single-turn NLU: intent classification and slot filling. Intent classification is the art of discerning the user's underlying intent, whether it be booking a flight or enquiring about the weather. Slot filling, conversely, involves identifying and extracting specific pieces of information from the input, such as dates, locations, or other pertinent entities necessary to fulfill the intent.

We commence with an exploration of the foundational techniques for intent classification. This journey takes us through traditional methods that harness dependency parsing and classical features, and then transitions to modern approaches employing machine learning models, word and character embeddings, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and advanced architectures like Siamese networks.

Subsequently, we venture into slot filling techniques, highlighting methods that span from classical machine learning models to sophisticated neural network-based approaches. Our coverage includes dependency parsing, classical feature extraction, named entity recognition (NER), context windows, delexicalization, CNNs, RNNs, pointer networks, and reinforcement learning strategies.

We then explore the domain of joint NLU techniques, which strive to simultaneously perform intent classification and slot filling. Joint NLU capitalizes on the

S. C. Han et al., *Natural Language Understanding in Conversational AI with Deep Learning*, https://doi.org/10.1007/978-3-031-74364-1_3

interdependencies between these tasks to enhance overall performance and robustness. Techniques in this category encompass classical methods, conditional random fields (CRFs), recursive neural networks (RecNNs), CNNs and RNNs, attention mechanisms, capsule networks, memory networks, and transformer models.

The chapter also investigates the role of external resources, such as knowledge bases, in augmenting NLU systems. These resources provide additional context and information that can significantly enhance the accuracy and relevance of intent detection and slot filling.

Finally, we survey a comprehensive array of datasets commonly employed in NLU research and development. These datasets traverse various domains, languages, and interaction types, offering a solid foundation for training and evaluating NLU models. Notable examples include the ATIS, SNIPS-NLU, FRAMES, TREC, Microsoft Cortana, and Facebook datasets, among others. We also shine a light on specialized datasets for multimodal and multilingual NLU tasks, such as SLURP and MASSIVE.

By the conclusion of this chapter, readers will possess a thorough understanding of the techniques and resources vital for crafting effective single-turn NLU systems. This knowledge will equip them to construct and refine dialogue systems capable of understanding and responding to user inputs accurately and elegantly.

3.2 Intent Classification Techniques

Intent detection is typically set up as a sentence classification problem. That is, a feature or features are constructed from the sentence, and these are passed through a classification algorithm to predict a class for the sentence from a predefined set of classes. As a classification problem, the techniques applied look to discover a well-defined decision boundary between the features. Intent classification differs from classification tasks in other fields due to the nature of the data which are text sentences, coming from spoken language utterances. Hence, at least initially, the features should look to capture semantic information in the sentence. Beyond the semantic information within the words, many approaches have been made to extend the feature set using internal (syntactic, word context) or external (meta-data, sentence context) information.

Major Areas of Research Research into intent classification in SLU has generally come from four areas: search engines, question-answering systems, dialogue systems, and text categorization. Early search engines applied text similarity to select the results for users. More recently, intent classification has been applied to understand the searcher's intent further, and this approach has been proven to give better search results. However, Web queries are usually short and informal, causing difficulties in classifying intents because of insufficient information. Similarly, answering questions from users also benefits from understanding the intent of questions to generate better-quality responses. In dialogue systems, it has been

shown to be useful to identify intents of users in order to give appropriate responses to users. Intent classification is also applied in more general NLP tasks, such as text classification, sentiment analysis, and scientific citation.

3.2.1 Overview of Technological Approaches

Before 2015, most papers focused on classical machine learning approaches, such as SVM, K-nearest neighbors, and random forest. Features used by these models were generated by dependency parsing, word embedding, and n-grams. Then, with the success of deep learning in other areas, neural networks, especially RNNs, started to be widely used for this task. Attention mechanisms have been integrated in models for identifying which parts of sentences should contribute to the classification. Since intent classification is proposed to be integrated with Web engineering, which requires the ability to understand short texts that contain less information, features used for training have been enriched by feature engineering using Web metadata.

3.2.2 Issues Addressed in Intent Classification

In this section, we survey the issues encountered in the literature around intent classification and the solutions proposed. The issues may be specific to the task, like ambiguity of semantic intent. They may be general machine learning issues like lack of training data and dealing with new or evolving domains. They may be issues specific to the available data like imbalanced data, short sentences, or the out-of-vocabulary (OOV) issue. Feature creation to capture information extra to that in the words is considered to boost performance. Extending the range of the task to multiple intents or to identify out-of-domain sentences is covered.

Ambiguity in Interpretation In essence, this issue is at the heart of intent classification, identifying the decision boundary between samples close together in feature space, yet belonging to different classes. This issue may be more prevalent with short texts, since they may include insufficient information and not follow correct grammar.

An early approach from [162] was to propose a rich feature representation with an ensemble learning framework giving different perspectives on the classification. The feature creation is covered further in ensuing sections.

In 2020, [180] proposed using contrastive learning. The model trains on triples of samples—an anchor sample, a positive sample in the same class, and a negative sample from a different class. Combining convolutional and BERT encodings of each one and mapping them to Euclidean space with Siamese shared weights, an intermediate loss of the anchor-positive distance minus the anchor-negative distance is minimized. The Euclidean mapping of the anchor is used for classification.

Lack of Labelled Training Data or Small Training Sets Collecting and labelling large amounts of data for training can be expensive. With small datasets, models are more likely to be overtrained. Further, the out-of-vocabulary (OOV) issue, where words appear in the test set that are not in the training set, is more likely to occur.

Sarikaya et al. [185] proposed a DBN-initialized neural network for intent classification to learn from unlabelled data and generate features for a feed-forward network. The feed-forward network is then fine-tuned on labelled data, which may be small in number but still give reasonable results. A DBN is a stack of Restricted Boltzmann Machines (RBMs). To train a DBN, the RBMs are trained layer-by-layer in sequence using parameters learned by previous layers. After training the stack of RBMs, the weights of the DBN are used to initialize the weights of a feed-forward neural network. This approach performed better than traditional machine learning models, such as maximum entropy and boosting, and is similar to SVM.

Hasanuzzaman et al. [78] tried to include temporal query understanding into Web search query intent classification. They tackled two major issues: one is the inadequacy of limited training data, while the other is the limited literal features able to be extracted from queries of short length (typically 3–4 words). They utilized external resources collected from the Web that may help bolster temporal information, such as Web snippets for queries and the most relevant year, date, etc. Based on this, 28 features were designed and extracted. They then proposed an ensemble learning solution framework defined as a multi-objective optimization problem (MOO) and explored it with 28 classifiers using different optimization strategies. The utilization of external resources and ensemble learning was intended to reduce bias to better handle the limited training data.

Methods for dealing with new unannotated domains and datasets by transfer of weights from models trained on annotated datasets combined with few shot training have been explored in the slot labelling and joint task area and are discussed later.

Sufficient data is essential for training a model. To work with unlabelled data, unsupervised training methods could be investigated in further research.

Multi-domain/Multi-lingual Generalizability Most text classification models focus on only one language, one domain, and also one task. To address this, [40] developed a novel multi-layer ensembling approach that ensembles both different model initialization and different model architectures to determine how multi-layer ensembling improves performance on multilingual intent classification. They constructed a CNN with character-level embedding and a bidirectional CNN with an attention mechanism. In addition, they explored LSTM and GRU with or without character-level embedding and attention mechanisms. When ensembling models, they use a majority vote approach.

Masumura et al. [137] proposed an adversarial training method for multi-task and multi-lingual joint modelling to improve performance on minority data. The language-specific network can be shared between multiple tasks, where words in the input utterance are converted into language-specific hidden representations. Next, each word representation is converted into a hidden representation that uses BiLSTMs to take neighboring word context information into account. Task-specific

networks can be shared between multiple languages, where the language-specific hidden representations are converted into task-specific hidden representations. The proposed method combines a language-specific task adversarial network with a task-specific language adversarial network.

Emerging Intents Detection In dynamic real-world applications, the intent set evolves. A method to detect and classify emerging intents is a desirable adjunct task.

Hashemi et al. [79] proposed to use a CNN to extract query features for intent classification, which is trained based on word2vec embeddings. They perform clustering on these query representations and observe that new examples far from the clusters could be used to identify emerging intents, though they do not perform that task.

Xia et al. [241] proposed two capsule-based architectures to detect emerging intents. They construct three capsules: SemanticCaps, DetectionCaps, and Zero-shot DetectionCaps. SemanticCaps is based on a bidirectional RNN with multiple self-attention heads and is used to extract semantic features from utterances. Then, DetectionCaps aggregates the low-level information from SemanticCaps to high-level information in an unsupervised routing-by-agreement approach to obtain intent representations. For detecting emerging intents, the Zero-shot DetectionCaps takes information from SemanticCaps and DetectionCaps to calculate vote vectors for information transferral. Then, the vote vectors are multiplied with similarity between embeddings of existing intents and emerging intents and summed to generate representations of emerging intent labels (Fig. 3.1).

Unseen Intents Dealing with intents that are unseen in the training data is a related challenging task. Lin and Xu [124] proposed a two-stage method to detect unseen intent labels. First, they used the output of a Bi-LSTM as sentence representation. In the second stage, the model takes this representation and applies a local outlier factor (LOF), a density-based detection algorithm, to detect unseen intents. The model uses large margin cosine loss (LMCL) as the loss function instead of cross-entropy, which aims to maximize the decision margin. In this way, the inter-class variance is maximized, and the intra-class variance is minimized. This is to ensure that the features extracted by the Bi-LSTM can be more discriminative (Fig. 3.2).

The OOV Issue Most approaches that perform their own learned word embedding are dependent on the vocabulary in the training set and may suffer from OOV issues, with small training datasets more affected. Using character n-grams is a common approach to handling unseen words based on the idea that similar words may come from a common root. Another is to replace all words below a chosen frequency in the training set with a special token, say UNKNOWN.

Ravuri and Stolcke [178] proposed character n-grams as their word input encoding method with both RNN and LSTM since they thought the OOV issues became more severe when using RNNs because an unknown word could propagate an effect to the consequent words. Similarly, [192] proposed sub-word semantic hashing for addressing the OOV issue, which comes with small training datasets. Before

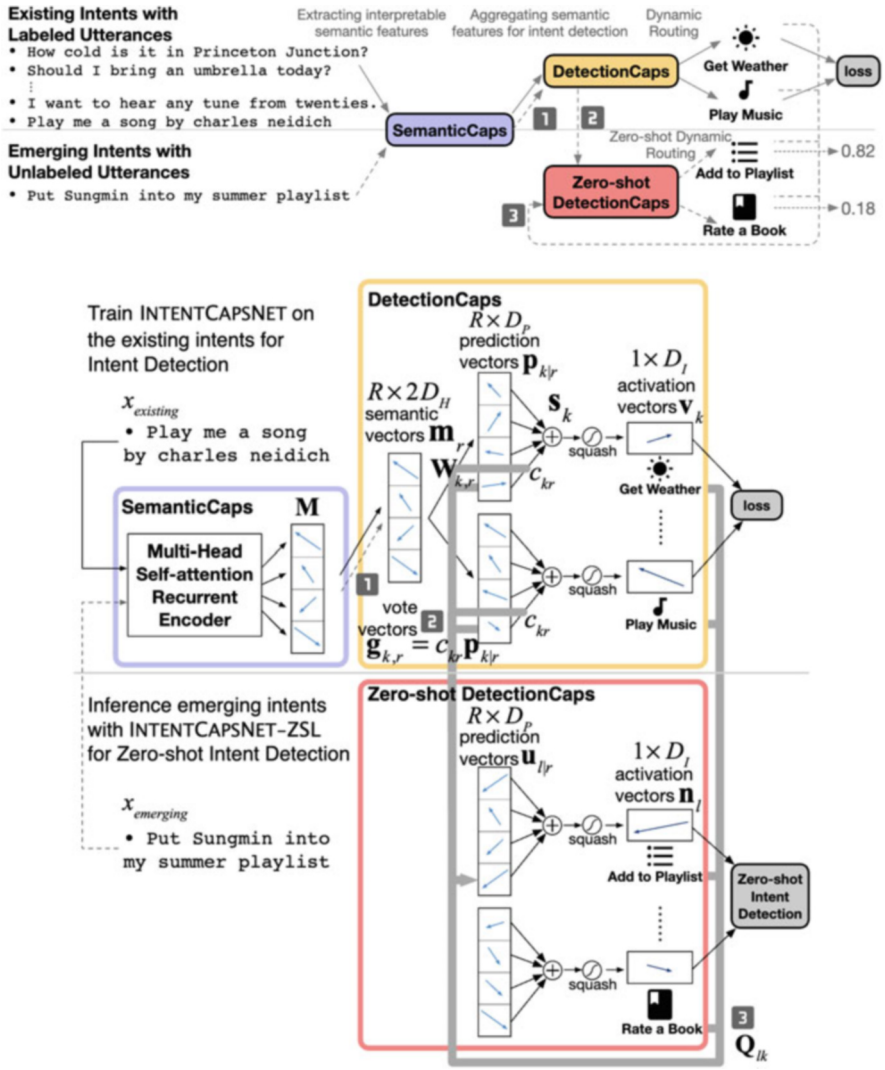

Fig. 3.1 The illustration of the proposed INTENTCAPSNET-ZSL model for zero-shot intent detection and its architecture details [241]

sub-word semantic hashing, sentences are transferred into lowercase, pronouns in sentences are replaced by "-PRON-", and special characters except stop characters are removed. Then, classes with fewer sentences are augmented by adding sentences generated using synonym replacement. After this, every token in each sentence is wrapped by two "#" symbols and represented using trigrams. These sub-tokens are then vectorized using an inverse document frequency vector and the Euclidean norm. In the end, the vectors can be used in any intent classification model.

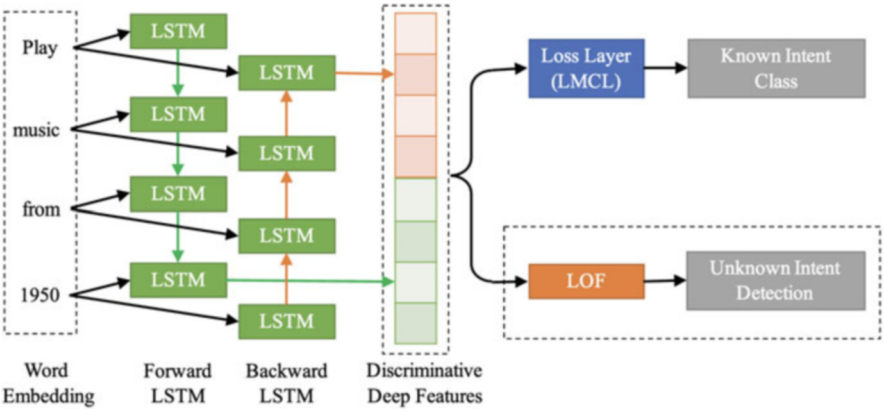

Fig. 3.2 The architecture of the two-stage method to detect unseen intent proposed by [124]

Short Text Queries User queries for search engines are usually short (3–4 words) and lack context, so it is essential to extract more information from queries for successful classification. Syntactic features (e.g., POS tags) and external knowledge sources have been used to enrich query features.

Hasanuzzaman et al. [78] included temporal query understanding into Web search query intent classification, and their model works well with the limited literal features for queries of short length. Purohit et al. [162] focused on intent understanding of social media text such as tweets. Some of these can be short, leading to ambiguity of interpretation and sparsity of relevant behaviors. They try to improve the expressiveness of data by utilizing multiple patterns from knowledge sources and fusing the top-down knowledge-guided patterns with bottom-up frequency-based representations for feature formation. Based on this, they utilize an ensemble learning strategy to reduce the bias.

Xie et al. [242] proposed a model called Semantic Tag-empowered User Intent Classification (ST-UIC) based on a constructed semantic tag repository. This model uses a combination of four kinds of features including characters, non-key-noun part-of-speech tags, target words, and semantic tags. After pre-processing, characters and target word features are extracted to maintain the contextual information. Then, key nouns are expanded using semantic tags, and POS tags are used as features if a query does not contain target words. With this approach, representation can be enriched for short queries.

In contrast, [214] tried to simplify the query input based on a dependency parser to generate simple and well-formed queries. They were motivated by the performance gain of existing statistical SLU models on simple, well-formed queries as well as the need to handle increased Web search queries formed by keywords. They simply kept the top-level predicate and its dependents for the query simplification and combined it with the sentence input for further classification using AdaBoost. The essence here is to try to provide the extracted keyword pieces

as auxiliary information, which proved to decrease the intent classification error rate. Because some semantic and syntactic information contained in the sentence is filtered out, when this simplified syntactic structure of a sentence was used alone as input for classification, a decrease in performance was reported.

This issue mainly occurs with user queries for Web searching. While not widely reported on in the joint task literature, short texts do occur in other datasets and the methods described above can be applied there. Rather than the rule-based feature construction approach, knowledge graphs may be further investigated as a method for adding relevant external information.

Other Feature Engineering Even when the utterances are longer, the challenge is to extract information relevant to the classification task. Many papers have used standard word embeddings, as well as RNN and CNN feature creation. In order to boost semantic understanding, [223] proposed to use both in a model called Character-CNN-BGRU. Firstly, this model uses character embeddings to represent sentences rather than word embeddings. A CNN takes the character embedding as the input and extracts local features via max-pooling after a convolutional layer. Meanwhile, a window feature sequence layer is added to the convolutional layer to obtain temporal information, which is important for the bidirectional-gated recurrent unit (BiGRU). Finally, the output of the max pooling layer and the output of BiGRU are concatenated and passed to a softmax layer. While semantic features are useful, methods to extract other usable information for classification have been developed.

Grammar Feature Exploration From the question-answering field comes a feature-creation technique based on grammar. Question words, such as "what," "how," and "where," and also domain-specific grammar may indicate the class of a question. Based on this idea, [146] proposed a grammar-based framework for question classification, which utilizes three features: grammatical features, domain-specific grammatical features, and grammatical patterns. Grammatical features are used to parse a question into a sequence of grammatical terms. Domain-specific features are used to identify the domains to which grammatical terms in the sentence correspond and tag them. After parsing and tagging each term in the question, the pattern is formulated. The classification task is processed using machine learning models, such as SVM or the J48 algorithm.

Imbalanced Data A dataset is unlikely to have the same number of samples for each class, and sometimes, the data can be quite imbalanced. Training a model using imbalanced data can cause poor performance in minority classes.

Purohit et al. [162] noticed that social media text corpora can have such imbalanced data. They suggested that two of their constructed features aid with imbalanced data in their dataset. These are Contrast Patterns features, where they mine sequential patterns within each intent class and then contrast them. Shridhar et al. [192] dealt with imbalanced data through oversampling by adding augmented sentences for classes with fewer samples during the pre-processing stage of their model.

ATIS, one of the major datasets for the joint task, is very imbalanced in the intent aspect, yet performance is excellent. The imbalanced data issue is rarely brought up in the joint task literature.

Co-occurrence of Words from Different Intents This is a particular form of ambiguity. Words important to different intents may co-occur in a query of a particular intent, and how these words are positioned may convey crucial information for intent detection in the current query.

Based on this idea, [264] proposed two types of heterogeneous information: (1) pairwise word feature correlations and (2) POS tags of the queries. The pairwise feature correlations are calculated based on cosine similarity between each semantic feature pair and learned by CNNs with pooling layers. Here, each dimension of the word vector is deemed to be a semantic feature. It tries to model the intent through these feature-level representations. Meanwhile, the POS tags provide the word-level information about word categories. Their experiment results show that utilizing the feature-level semantic representation outperforms the baseline model using only word-level features, and incorporating POS information with the feature-level representation significantly improves the performance.

Contextual/Temporal Information Modelling A single sentence in a conversation can be ambiguous, but the ambiguity can be eliminated if previous utterances are considered during intent classification. Meanwhile, Web queries are sensitive to time, and the intent carried by them may change over time, for example, as world events wax and wane in importance. Therefore, some studies incorporate contextual and temporal information in intent classification.

With multi-turn dialogue, [15] included the context from previous queries for the intent classification and slot filling of the current query. Each sub-task is treated separately, so this is not a joint model. For intent classification, they compared two approaches. The first constructs a context session feature by simply using 1 for the type of intent inferred from the previous utterance set and 0 for all others. This is combined with a basic bag-of-n-gram feature for the current utterance and fed to an SVM for intent classification. The second approach is to treat the query sequence as a sequence labelling problem using SVM-HMMs with a Viterbi algorithm. The incorporation of intent from previous utterances as additional information showed a significant reduction in error rate.

Hasanuzzaman et al. [78] utilized external resources collected from the Web that may help bolster temporal information, such as Web snippets for queries and the most relevant year, date, and other time indicators. Based on this, features are designed and extracted. Another solution to generate features is referring to one of a set of contemporaneous events, known as event-based Web searching. Kanhabua et al. [96] utilized the event-related log patterns that reveal both implicit and explicit temporal information needs, together with general lexical information such as named entities for feature representation. Classification is performed by various classical and neural methods.

Chen et al. [31] included temporal information in their model as well. They proposed a metadata feature for enhancing community query intent classification. The

metadata feature included query topic, query time, and user experience indicated by the number of previous queries. They firstly explored supervised learning using the text and metadata features separately. The result showed that using both features together outperformed the separate experiment for query intent classification. This led them to use a co-training procedure, which is a semi-supervised learning framework that can utilize a small amount of annotated queries plus a large amount of unlabelled ones. During the training, two separate and independent classifiers are trained first based on the two features. Then the prediction with higher confidence from either of the classifiers will be used to label the unlabelled query for training until a stopping criteria is reached. Experimenting with SVM, higher micro and macro F1 scores were achieved from semi-supervised co-training compared to supervised learning with a combination of the two features.

Rather than temporal information, [167] proposed the construction of multiple features from user metadata, regex extraction of named entities, and probabilistic context-free grammar of composite entities.

Contextual and temporal information may not be suitable for all datasets, but when available, future research can attempt to use it with their models. The multi-turn dialogue approach will be explored further under the joint task. Graphs may also be explored for integration of contemporaneous topics and events.

Target Variation Typically, in intent classification, the predefined intents are enumerated, and a cross-entropy loss is calculated. Qiu et al. [167] rather calculated LSTM embeddings of the training sentences and averaged them within samples of the same intent. For test prediction, they calculate a similarity measure of the LSTM embedded test sample with the training samples averaged by intent and choose the closest.

Generalizability via Ensembles Overfitting of models to the training data distribution leads to poor generalizability. One method to address this is to use ensembles of models to synergistically exploit the benefits of each. Deep learning architectures, such as LSTM, GRU and CNN, have been used frequently in intent classification. Firdaus et al. [57] proposed to combine those deep learning architectures. To start exploring combinations of different models, they constructed CNN, LSTM, and GRU individually. Then, four ensemble models were built, which are CNN-LSTM, CNN-GRU, LSTM-GRU, and CNN-LSTM-GRU. In these models, predictions from individual models are combined using an MLP model. Qiu et al. [167] also used an ensemble of classical methods, being random forest, SVM, and Naïve Bayes and softmax regression, with the ensemble outperforming the components.

Out-of-Domain Utterances Not all utterances made to SLU devices contain an intent related to the purpose of the device. They may be incidental conversations or have an intent that the device is not meant to fulfill. Kim and Kim [100] augmented a dataset with such utterances and performed a multi-task learning approach to perform the classification of in-domain utterances and detection of out-of-domain (OOD) utterances simultaneously. A loss function, which maximizes the intent accuracy while accepting a value of false acceptance rate (1-recall) for OOD

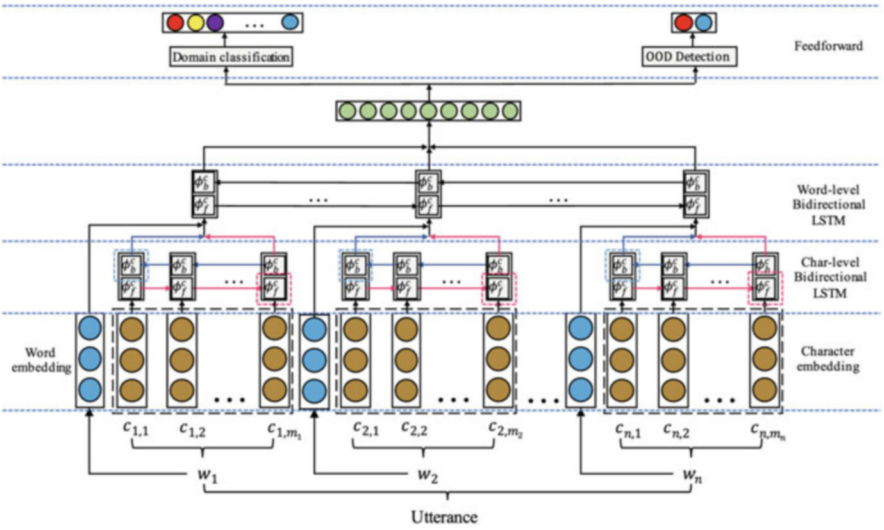

Fig. 3.3 Model architecture proposed by [100]: each word is represented by the concatenation of the word vector from word embedding and the orthography-sensitive vector from the last outputs of two character LSTMs

utterances below a threshold, is back-propagated. Including the second task boosts the intent detection performance (Fig. 3.3).

Yilmaz et al. [255] also constructed augmented datasets with OOD utterances. They constructed a vector of KL divergence values for subsequent pairs of intent probabilities determined from hidden states of a unidirectional LSTM fed with word embeddings. The KL vectors are fed to an SVM and Naïve Bayes and Logistic Regression for OOD classification. Logistic regression gives the best results.

Multifaceted Query Intent Prediction Most annotated queries in training datasets express only one intent. In real life, however, queries may contain more than one intent. For example, the query "find Beyonce's movie and music" has two intents, 'find_movie' and 'find_music'.

One straightforward solution is to use the top couple of predictions from existing single-label classifiers. Another solution is having binary classifiers for every label in single classifiers. González-Caro and Baeza-Yates [66] used this approach. In their model, each utterance has multiple facets, each a class with at least two categorical labels. For instance, in the Task facet, the intent can be "Informational," "Not Informational," or "Ambiguous." They experimented with linear SVMs for intent classification of different combinations of the facets. Compared with the corresponding single-facet classification, additional supervision from multiple labels led to improvement of the overall performance. This additional information was found beneficial to small categories (classes with few samples in the corpus), with the recall of those small categories improving.

Treating multi-labels as atomic labels has been explored in some studies. This approach may suffer from a data sparsity problem but has good classification accuracy. Based on this, [246] proposed two approaches to exploit the information shared among different intent combinations. The first one is adding class features, which is to add n-gram features for combined intent appropriate to the separated intents. For example, the multi-intent label, 'buy_game#play_game', should have added two features for the embedded intents 'buy_game' and 'play_game'. The other is adding hidden variables to identify segments belonging to each intent. Instead of using existing segmentation algorithms, they add a layer of hidden states corresponding to each word. This can indicate the embedded intent to which a word most strongly aligns. Following that, a perceptron layer performs the classification.

Understanding queries with multiple intents can make conversations with dialogue systems more natural and smooth, and there is recent research on this in the joint task. Further research could explore more approaches to modelling relations among label combinations.

3.3 Slot-Filling Techniques

Slot filling is the second critical task in natural language understanding. It is the attachment of a label to each token in an utterance. The label describes the type of semantic information contained in the word represented by the token. Slot filling is a sequence labelling task. The task is learning not just slot label distributions for words but also what slot labels typically co-occur in utterances (label dependency) and in what order. Reference to context in both directions of a word should be included to maximize performance.

Major Areas of Research Slot filling is a key task in dialogue systems, to interpret natural language from users, from which the system can judge what information to retrieve or what task to complete for the user. Slot-filling models are also integrated with online shopping Web sites, whose core is a task-oriented dialogue system. Product search queries can be better understood, and shopping assistance can be provided to customers. The field of question-answering has provided research to extract the semantic features within queries.

3.3.1 Overview of Technological Approaches

Traditionally, generative models (for example, HMMs [228]), which capture the joint probability distribution of the utterance tokens and their slot labels, and discriminative models (like CRFs), that estimate slot label conditional probabilities given the observed token sequence, were used to address sequence labelling

tasks. With the success of deep learning, researchers experimented with putting components, for example, CRFs and brief networks, within a deep structure.

Since 2013, RNNs have been increasingly popular in this field. In RNNs, each word can access information from the previous context words. Bidirectional RNNs are applied to utilize both the past and future context. However, the distance between words is linear in RNNs, and the vanishing gradient problem may occur, meaning that long-term dependencies cannot be learned by the model. As a result, LSTM cells began to be used more frequently because of their ability to forget unimportant information and more successfully model longer dependencies. However, even LSTMs can underperform with very long sentences. Other models are thus incorporated to capture label dependency, which refers to the situation that some slots appear in context with some other slots more frequently, and perform sequence-level optimization. An architecture with CRF layers sitting above neural networks is often used to address label dependency directly.

Another issue is that RNNs are typically not able to process multiple words simultaneously due to their sequential nature. Thus, attention mechanisms that can take more tokens into account simultaneously than RNNs are used. Additional features are also incorporated to improve the performance of RNNs, such as named entity features, segment features, and external memory. Integrating extra knowledge from different sources is also considered an effective approach. Some recent slot-filling models also attempt to handle unseen semantic labels and multiple domain tasks by adapting neural CRFs and label embedding.

Slot-Filling Architectures Models that have proved reliable for sequence labelling in other fields have been adopted to address the slot-filling task. RNNs with different architectures have been explored in many studies [142, 143, 219, 253], considering their promising performance in sequence modelling elsewhere.

CNNs have also been used. Vu [218] proposed a bidirectional sequential CNN for slot labelling, which considers both previous contextual words with preserved order, surrounding context, and also past and future information. To label a word, two matrices are separately generated for the previous and future context words of the word. These are combined to form a matrix for the current word. Then, the two matrices for the past and future information are passed to corresponding vanilla sequential CNNs. The output of networks is concatenated with the matrix for the current word. After that, two matrices can be combined using a weighted sum of the forward and the backward hidden layer or concatenating. For training, a ranking loss function is applied.

Korpusik et al. [108] performs a useful comparison of BiGRU-, CNN-, and BERT-based models with the then new BERT outperforming the other candidates (Fig. 3.4).

A summary of slot-filling techniques appears in Table 3.1. The incorporation of newer technologies for slot labelling, particularly attention-based methods like the transformer, has been subsumed by the joint task in more recent years, as will be explored in Sect. 3.4.

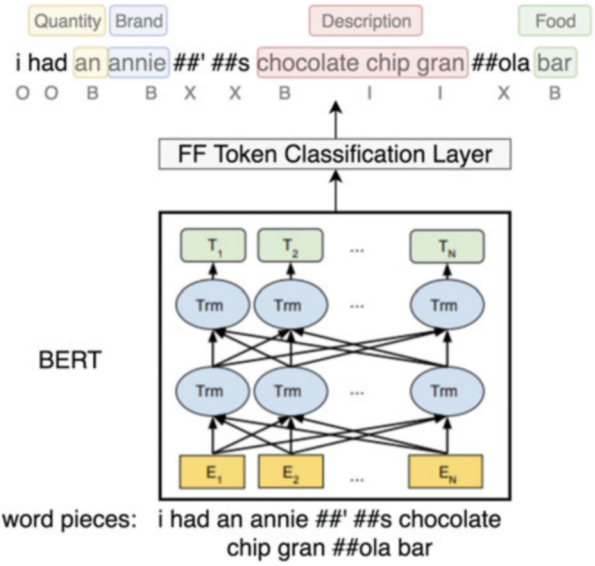

Fig. 3.4 An illustration of how BERT is used to generate contextualized word embeddings, which are then fed into a finetuned linear token classification layer on top [108]

3.3.2 Issues Addressed in Slot Filling

Like intent detection, issues found in slot filling can be task specific, more general machine learning issues, dataset issues, or may involve feature engineering or methodological approaches for better results. Slot filling introduces issues typical to sequence labelling tasks like label dependency and long-distance dependency.

Label Dependency There are dependencies between slot labels, meaning that some slots appear more commonly with some other slots in the same utterance. For example, in a travel dataset, it is probable that *B-FromCity* and *B-ToCity* are in the same sentence. Capturing such label dependencies would help find the best slot combinations and generate better prediction results.

One approach is to integrate CRFs and regression models. Yao et al. [251] applied an LSTM for the slot-labelling task and included a regression model to capture label dependencies. Yao et al. [252] proposed a recurrent conditional random field (R-CRF), which integrates recurrent neural networks and a CRF to explicitly model the dependencies between semantic labels and achieve sequence-level optimization. Another solution utilizes the encoder-decoder architecture, which had been applied for machine translation previously. The input sentence is encoded by an LSTM and is used as the input of another LSTM, which considers label dependencies [112, 125, 280].

Table 3.1 Slot-filling papers reviewed with addressed issue and approach

Paper	Addressed issue	Approach
[258]	Long-range state dependency	Deep learning, CRF
[46]	To extend DCN	Kernel learning, deep learning, DCN, log-linear model
[47]	Data sparsity problem (CRF)	Deep belief network (DBN)
[143]	To explore RNN	RNN
[253]	To explore RNN-LM	RNN-LM
[252]	Label dependencies, label bias problem	RNN, CRF
[251]	Gradient diminishing and exploding problem, label dependencies, label bias problem	LSTM, regression model, deep learning
[125]	Label dependencies	RNN, sampling approach
[142]	To explore RNN	RNN
[160]	Vanishing and exploding gradient	RNN, external memory
[112]	Label dependencies	LSTM, encoder-labeler
[219]	To explore past and future information	bidirectional RNN, ranking loss function
[218]	To explore CNN	CNN
[280]	To explore attention mechanism	Bidirectional LSTM, LSTM, encoder-decoder, focus mechanism
[45]	Unseen slots	CRF
[65]	To explore MTL	MTL, segment tagging, NER
[129]	To explore MTL	MTL, NER, bi-LSTM, CRF
[191]	To better labelling common words	Encoder-decoder attention, delexicalized sentence generation
[224]	Imbalanced data	DNN, reinforcement learning
[276]	OOV	GRU, attention, pointer network
[65]	To explore MTL	MTL, segment tagging, NER
[101]	To extend original SLU to H2H conversations	Bi-LSTM, different knowledge sources
[189]	Continual learning	Bi-LSTM, context gate
[217]	Restricted utilizing contextual information	MTL, Bi-LSTM
[108]	Architecture comparison	BiGRU, CNN, BERT
[130]	Low-resource dataset	MTL
[281]	Data sparsity problem	Prior knowledge-driven label embedding, CRF
[273]	Long-range dependency, vanishing gradient	TDNN

Long Range Dependence Long-range dependence refers to the challenge of capturing the importance to each other of data points far apart in space or time where there is a decay of statistical dependence over distance. While the short text queries described in the intent section should not suffer from this, utterances in SLU datasets can be much longer.

The approach from [258] is a deep network of layered CRFs. Lower layers can generate frame-level marginal posterior probabilities. Then the higher layer takes these probabilities along with the observation sequence of the previous layer. In the end, the highest level will make the final predictions. Zhang et al. [273] addressed the shortcomings of RNNs, being the long-range dependence issue and vanishing or exploding gradient. They used a deep stacking of time delay neural networks for feature creation. These create convolutional features from context windows of varying sizes and varying step sizes through the sentences. The features are then passed through a final RNN and classified.

The transformer architecture, which utilizes self-attention across utterances, and memory networks, which can store longer range information, are now studied in the joint task, as described in the next section. Focused research using these methods for only the slot task may uncover useful methods for use in the joint task.

The Label Bias Problem Label bias is a sequence labelling issue related to maximum entropy Markov models (MEMM). In MEMM, the states with a single outgoing transition can ignore their observation, meaning that the model can tend to stay in a state which is unlikely to happen. Most approaches to this problem combine CRFs with RNNs.

Yao et al. [251] applied LSTM cells, which contain a gate which can forget unimportant information, and also incorporated a regression model to model label dependencies. In order to avoid the label bias problem, the regression model took non-normalized scores before softmax. The R-CRF of [252] also addresses label bias. Similarly, [142] explored RNNs with multiple different architectures and proposed to apply Viterbi encodings and recurrent CRFs to eliminate the label bias problem.

Learning Common Words The surrounding words of one slot in different sentences are usually similar. For example, the word "to" is highly likely to lie between slots labelled *B-FromCity* and *B-ToCity*. Therefore, [112] thought that learning the common words around slots in different ways may be helpful for slot filling. Shin et al. [191] introduced a model which can jointly generate delexicalized sentences and predict labels using an encoder-decoder framework with input alignment. In the delexicalized sentences, words are replaced by their corresponding slot labels. For example, to delexicalize the sentence "i want to fly from baltimore to dallas," the words "baltimore" and "dallas" should be replaced by *B-FromCity* and *B-ToCity*. This approach is based on the fact that different words that correspond to the same slot usually play a similar semantic and syntactic role in the sentence, which allows the model to learn the common words surrounding the slots.

Low-Resource Datasets A more general machine learning issue is that methods generally rely on the presence of annotated training data, which is costly to produce. Louvan and Magnini [130] tested models which also trained on a large freely available annotated dataset for a similar task (NER for example) in a multi-task learning environment (see Sect. 3.3.2. They then used 10% of the available training data from slot tagging datasets to train the slot-labelling task. They varied this

proportion and observed when it reached 40% that the auxiliary task stopped having much effect.

Diminishing and Exploding Gradients In neural networks with multiple layers, or long RNN sequences, the gradient may become vanishingly small, preventing the weights from changing value during back-propagation. Alternatively, large gradients may self-propagate and lead to unstable networks. Yao et al. [251] combined an LSTM and a regression model for slot filling. LSTM are partially designed to address these issues. To avoid the gradient diminishing and exploding problem, the memory cells within them are linearly activated and propagated between different time steps.

Peng and Yao [160] introduced an external memory to overcome the limitations on memory capacity of simple RNNs, and therefore, the diminishing and exploding gradient problems can be addressed. The hidden layer has an additional input that comes from the external memory. A weight vector is used to retrieve the content from the external memory, which is determined according to the similarity between the contents and the hidden layer.

Data Sparsity Problem Data sparsity can be an issue when there are a large number of discrete features. If those discrete features are represented using matrices, the matrices can be large and sparse, which may lead to a model ignoring the relations among features. Zhu et al. [281] noticed that the data sparsity problem can also occur with labels because labels are sometimes encoded using one-hot vectors. They proposed a label embedding that is constructed using prior knowledge, including atomic concepts, slot descriptions, and slot exemplars. An atomic concept assumes that each slot can be represented as a set of atoms. Slot descriptions are the textual descriptions for slots in natural language. Slot exemplars are to extract label embeddings for slot labels, which contain values of each slot and their neighbor contexts (Fig. 3.5).

Continuous Learning With new data becoming available quickly, it is desirable to have a retrained model incorporating the new data. However, the training process

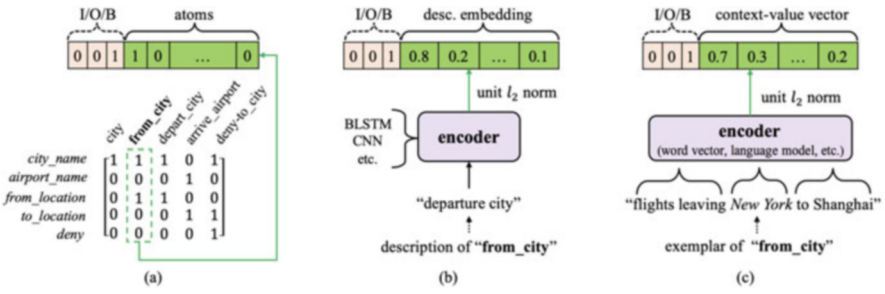

Fig. 3.5 Three different label embeddings: (**a**) atomic concept, (**b**) slot description encoding, (**c**) slot exemplar encoding—consisted of IOB symbol and slot vector proposed in [281]

could be time and cost consuming, and keeping all the existing data can introduce redundancy. Thus, there is a problem of how to continuously learn from new data.

Shen et al. [189] proposed a ProgModel, consisting of a context gate. This gate aims to transfer previously learned knowledge to a small expanded component, which is placed after the hidden state of the new model. The training procedure is progressively conducted at each batch. Therefore, the model can learn faster from the new training data without catastrophically forgetting the previous expressions.

Imbalanced Data Training datasets can be imbalanced, dominated by some tags while only containing a small number of examples of other tags. This can lead to poorer performance in the minority tags. One solution from the slot-labelling literature comes from [224] who design a deep reinforcement learning (DRL) based augmented tagger with a deep neural network. While the whole dataset is used in the training part, only partial data with unsatisfactory performance will be evaluated by the augmented tagger.

Unseen Labels As with intent classification models, slot-labelling models can rely heavily on the training data and will struggle to correctly assign a label which does not appear in the training data.

Zhao and Feng [276] proposed a model with a pointer network to solve this problem. This model predicts slot values by jointly learning to copy a word which may be out-of-vocabulary (OOV) from an input utterance through a pointer network, or generate a word within the vocabulary through a seq2seq model with attention (Fig. 3.6).

Dai et al. [45] proposed an elastic conditional random field (eCRF) which can utilize semantic meaning in slot embedding for open-ontology slot filling. The

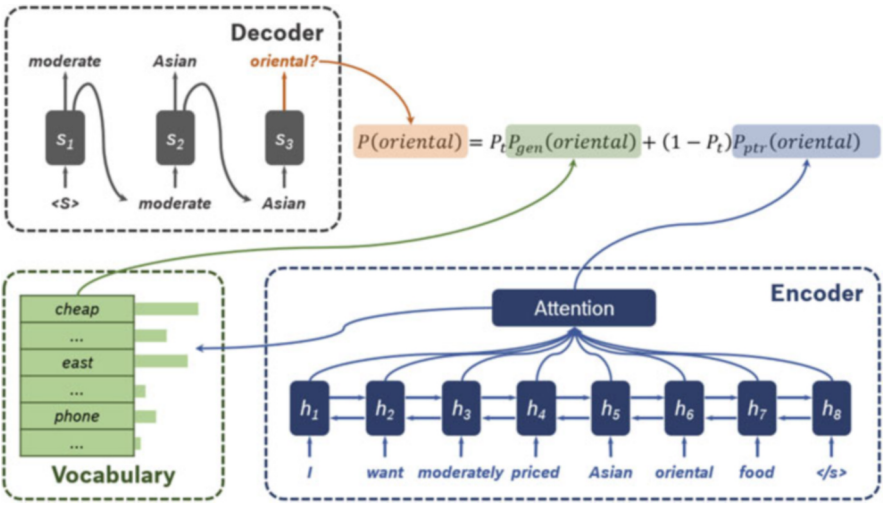

Fig. 3.6 Zhao and Feng [276]'s model for slot value prediction based on Seq2Seq learning with attention and Ptr-Net

model has a slot description encoder which takes all slot descriptions as input, and outputs distributed representations for slots. Meanwhile, a BiLSTM is used to extract features from utterances. Then, the eCRF labeller, which is a potential function containing two terms for semantic similarity of the slot descriptions and the extracted contextual features and interactions between the slot labels, is applied.

Exploring Multi-task Learning Multi-task learning (MTL) is the idea that similar auxiliary tasks can assist a main task. Some papers have used this idea, setting slot labelling as the main task.

Louvan and Magnini [129] attempted to perform slot filling and named entity recognition (NER) jointly in a multi-task framework considering that most slot values are also named entities and NER has high state-of-the-art performance. They mentioned that a better slot tagging result can be achieved if NER is at a lower level. Similarly, [65] investigated hierarchical multi-task learning to perform low-level tasks first, namely named entity tagging and segment tagging, and then the high-level task, that is slot labelling, can make use of the results from the low levels with cascade and residual connections. Louvan and Magnini [130] then performed a more wide-ranging experiment with two auxiliary tasks (NER and semantic tagging (SemTag)), and a comparison of a training on auxiliary task(s) followed by fine-tuning on the main task approach versus the previously explored hierarchical approach. They found generally the hierarchical approach gave better results and that parsimoniously using only one auxiliary task (NER) worked better.

Most models utilize contextual information, but they use it in a restricted manner, for example, self-attention. Therefore, [217] proposed a multi-task setting to train a model to incorporate the contextual information at two different levels, which are representation level and task-specific level. This multi-task setting includes the slot filling as the main task and two auxiliary tasks. The first one is to increase consistency between the word representation and its context, and another one is to enhance task-specific information in contextual information.

Extending to Human-to-Human Conversations Task-oriented language understanding in human-to-machine (H2M) conversations has been extensively studied. An interesting twist is to perform slot tagging in human-to-human (H2H) conversations. Here, the agent is a third party listening in, not being directly asked to perform a task.

Kim et al. [101] focused on slot filling in H2H conversations and explored LSTMs with different knowledge sources. First, the character embedding and the word embedding are concatenated for each word. Then, the embeddings are passed to a bidirectional LSTM, which will be used for making final predictions. Furthermore, there is an additional model which can utilize knowledge from multiple sources, including sentence embeddings for H2H conversation, contextual information, and H2M expert feedback. The sentence embeddings are generated by a sentence-level embedding model trained using tweets with URL and Web search queries to the same URL. The contextual information is extracted from previous utterances in the conversation. Also, this model uses pre-trained slot-filling models for H2M conversations on similar domains as the expert model. The knowledge

from three sources is encoded into vectors which are then combined and aggregated with the output of the bidirectional LSTM.

Slot tagging introduces the problem of generating a sequence of hidden labels to a sequence of word tokens, qualitatively different to the intent classification task. Together, in SLU, the two sub-tasks contribute to a better representation of the semantics of an utterance than each one separately. In the next section, we consider the joint task where both are addressed in one model.

3.4 Joint NLU Techniques

The joint task integrates the objectives of the two sub-tasks. Many papers highlight the relationship between slot labels and intent, suggesting that a model should learn the joint distributions of these elements. Additionally, the model should consider the distributions of slot labels within utterances, drawing from the slot-labelling sub-task's dependency approaches. Methods for the joint task range from implicitly learning the distribution, to explicitly learning the conditional distribution of slot labels over the intent label and vice versa, to fully explicit learning of the entire joint distribution. The joint task also inherits the issues from each sub-task. Currently, the bulk of research focuses on the joint task, introducing newer methods not previously applied to the individual sub-tasks.

Major Areas of Research Research on the joint task has primarily emerged from the personal assistant and chatbot fields. Chatbots are typically task-oriented within a single domain, while personal assistants may operate across single or multiple domains. Other contributing areas include IoT and robotic instruction (where "intent" may refer to the robot's intended action), as well as in-vehicle dialogue for autonomous vehicles. These fields also require filtering out irrelevant utterances. Researchers have used data from question answering systems. Multi-turn dialogue datasets come from discussion between users and agents, sometimes artificially generated.

3.4.1 Overview of Technological Approaches

In this section, we give a detailed description of architectural and technical approaches used to address the joint task. In Fig. 3.7, we give a taxonomy of these approaches, breaking into feature engineering methods and architectural responses.

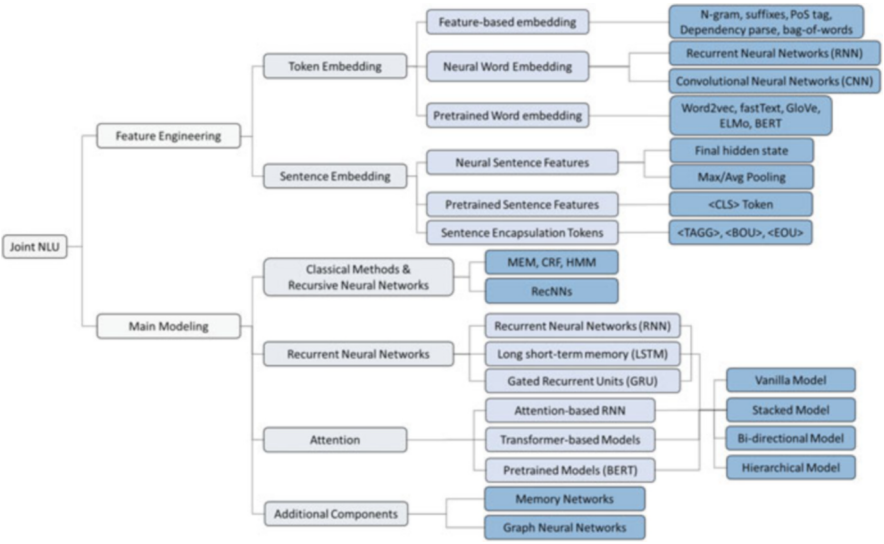

Fig. 3.7 Summary of technological approaches in joint NLU

3.4.1.1 Classical Methods

Perhaps the first model to show that the two tasks handled simultaneously performed better than in a sequential pipeline was described in [91]. This used a multi-level CRF. Other statistical models considered were a maximum entropy model (MEM) [227] and a HMM [28]. Although such classical models were rapidly overtaken in the boom of deep learning models from 2013 on, they still appear. This is particularly true of CRFs with their superior handling of label dependence as discussed in the previous section.

3.4.1.2 Recursive Neural Networks

The first neural model to address the joint task used recursive neural networks (RecNNs) [71]. RecNNs work over the constituency parse tree of an utterance, with leaves corresponding to the word vectors. The network computes a state for each node, then for each node, the states from child nodes are combined with a weight vector representing the node's syntactic type. A slot label classifier for each leaf is informed by the word vectors of itself and its neighbors and the state vectors along the path from the leaf to the root. The state at the root informs intent classification. The results were below the then state of the art for the tasks treated separately, but the gate was opened.

Since 2015, RNNs have begun a period of favor as their performance in sequence labelling tasks has been waxing. In the standard encoder-only approach, token features are passed in temporal sequence to the RNN. The RNN's hidden states at each step inform the slot labelling, and the final hidden state, when the network has consumed all the tokens in the input sentence, is used for intent classification. Bidirectional RNNs (BiRNNs), where the token sequence is considered in both a forward and backward direction, allow word context in each direction to contribute to the predictions.

The encoder is often extended to become an encoder-decoder. The decoder is a second RNN that produces a sequential output, for example, the slot label prediction for the current token. Liu and Lane [126] employed three variations for the joint task, as illustrated in Fig. 3.8. On the left hand side of each, the bidirectional RNN encoder first encodes the utterance into a sequence of hidden states h_i. The final hidden state is used for intent prediction and is also an input to the slot label decoder.

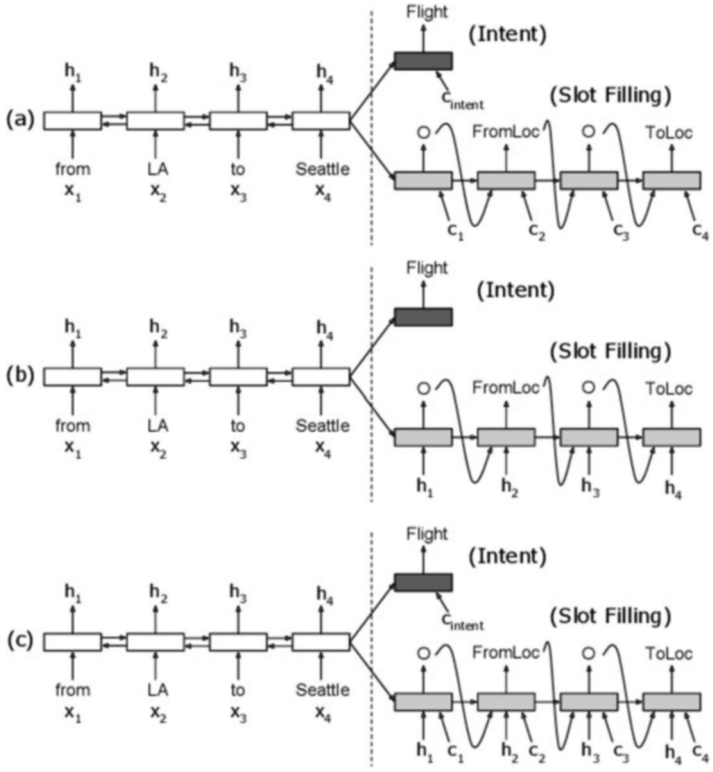

Fig. 3.8 Encoder-decoder model for joint intent detection and slot filling, proposed by [126]

Learned weighted sums of the hidden states are named the context representations, c_i. The variations in the paper differ with respect to the inputs to the decoder RNN at each time step, being the previous steps prediction and then some combination of the h_i and c_i terms. In this way, the paper was able to explore what encoded information was useful to the decoder process.

LSTM and GRU cell designs were quickly incorporated as they could address issues with long-term dependence. Zhou et al. [279] studied a two-layer LSTM architecture and [234] made an extended study of multi-layer LSTMs using different parts of the network to predict slots and intent. A joint loss, typically a sum of the cross entropy slot and intent losses, was used, but [277] kept the losses separate for back propagation through their RNN network. The prediction is typically a softmax on the decoder hidden state at each step, but [56] added a multilayer perceptron (MLPs) between the RNN unit and the softmax. As an alternative to predicting a slot label at each step, [105] make a global slot prediction from a matrix of the hidden states to a matrix of slot tag probabilities for each word (Fig. 3.9).

For RNNs, the input is typically token by token. Hakkani-Tür et al. [74] compared that to using context windows, where a sliding window of tokens is used at each time step, with superior results. They also used a special sentence token at

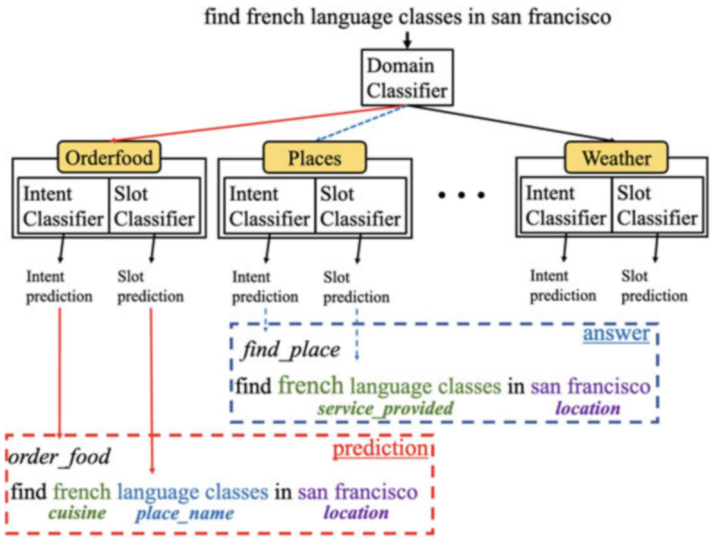

Fig. 3.9 Illustration of the pipelined procedure proposed by [105]

the end of the utterance. Zheng et al. [277] also used context windows, but their results showed a decrease in performance metrics.

Attention and Explicit Exchange In the RNN circuits described so far, the exchange of information between the two tasks is implicit, typically via a shared encoder or a joint loss. It was proposed that the internal circuit representations used for one task should be shared explicitly with the other task to strengthen the learning of the joint distribution of slot and intent labels. Attention is one way of forming the learned representation of one task's intermediate steps for delivery to the other task.

We have already introduced a form of such attention with the context representations c_i and c_{intent} in Fig. 3.8 from [126]. These are separately learned weighted sums of the encoder's hidden states. However, these still don't deliver information from one task to the other. Goo et al. [67] proposed an explicit attention exchange (Fig. 3.10). Again they use a BiLSTM and produce context representations for each token and for the intent (Equation 3.1). Their innovation is the slot gate g in Equation 3.2, which combines the intent context representation with the current slot context representation at each step. Then in Equation 3.3 the gate output is combined with the current hidden state and the current context representation to inform slot label prediction, y_i^S, for the current token. In the equations, h_i, h_j are BiLSTM hidden states, the α terms are learned attention weights, v is a learned vector, and W and W_{hy}^S are trainable matrices. Because in this circuit, we are feeding intent prediction information to slot prediction, and we type this circuit as *intent2slot*. Other versions of slot2intent were explored [118, 164, 221, 270, 275].

$$c_i^S = \sum_{j=1}^{T} \alpha_{i,j}^S h_j, c^I = \sum_{j=1}^{T} \alpha_j^I h_j \tag{3.1}$$

$$g = \sum v \cdot \tanh(c_i^S + W \cdot c^I) \tag{3.2}$$

$$y_i^S = \text{softmax}(W_{hy}^S(h_i + c_i^S \cdot g)) \tag{3.3}$$

A slot2intent approach was then described by [260] using a learned weighted sum of a BiLSTM encoder's hidden states and a weighted sum of the predicted slot labels.

Moving on from weighted sum context representations we next encounter self-attention. In self-attention, each element of a sequence calculates an attention score with each other element of the sequence. In [117], two self-attention layers exist. In the first attention is calculated on the word embeddings and convolutional character embeddings (Fig. 3.11). These attention scores are concatenated with the word vectors and fed to a BiLSTM layer. Dot product self-attention takes place on the BiLSTM hidden states $H = [h_1, ..., h_T]$, as is shown in Eq. 3.4.

The final hidden state passed through an MLP, producing a vector v^{int}. This is used for intent prediction and is also combined with each self-attention score produced from the BiLSTM hidden states (Eq. 3.5), and these are then combined

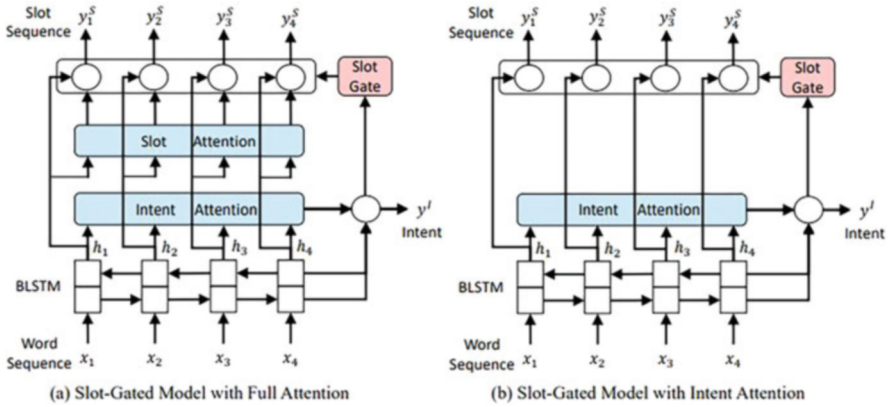

Fig. 3.10 The architecture of the proposed slot-gated models [67]

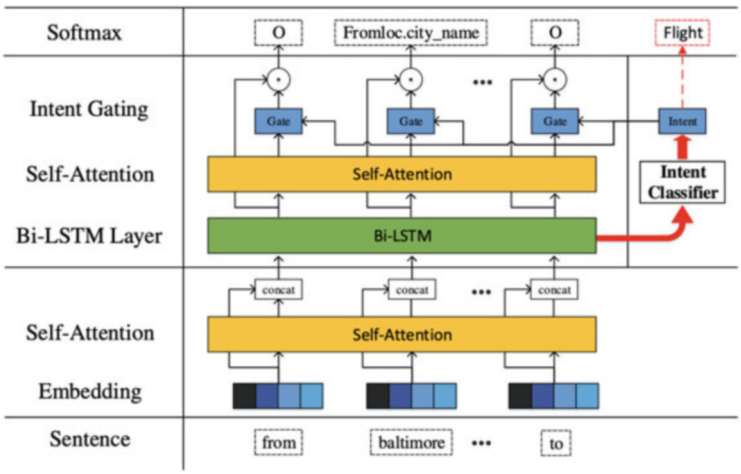

Fig. 3.11 Illustration of a self-attention model with gate mechanism for joint intent detection and slot filling proposed by [117]

with the hidden states themselves for slot label prediction (Eq. 3.6). Note that this is intent2slot flow only.

$$s_t = \text{self-attention}(h_t, H), H = [h_1, ..., h_T] \tag{3.4}$$

$$h_t^* = \text{MLP}([s_t, v^{int}]) \tag{3.5}$$

$$o_t = h_t \odot h_t^* \tag{3.6}$$

Chen et al. [32] also uses word and character embeddings but performs multi-head self-attention. In multi-head self-attention, multiple different self-attention

calculations are calculated and then recombined. In this architecture, they are fed to a BiLSTM with the final state informing intent detection, and the hidden states going through another multi-head self-attention before entering a CRF for the slot labelling.

Xu et al. [244] introduced a length variable attention. For each sequence location, a weighted sum attention of a sub-sequence of hidden states around the current token is calculated. The width of the sub-sequence is learned and takes the current location's hidden state as an input.

3.4.1.3 The Transformer

The now ubiquitous transformer architecture [216] would seem to have an in-built ability to handle the parameters of the NLU task. While non-recurrent, its strength is in capturing global, long dependence via multi-layer, multi-head self-attention. In the transformer encoder, the production of a sequence of deep contextual representations of the same length as the input sequence would seem well suited to the slot-labelling task. In fact, one of the first applications of the transformer encoder unit was in [209] who only used it for intent detection, passing the concatenated output embeddings from a single-layer transformer through feed forward layers for the intent prediction. Zhang and Wang [268] instead used a three-level transformer encoder and passed the output sequence to a CRF for the slot sub-task. They used a start-of-sentence token's output representation for the intent detection.

However, even with a CRF, it was observed label dependence issues were occurring, possibly because the use of positional encoding in the transformer is not as strong as the positional guidance in an RNN architecture. The transformer, initially designed for translation tasks, comes with a decoder for concentration on meaningful sequential output. Cheng et al. [35] added back the decoder for sequential slot label generation.

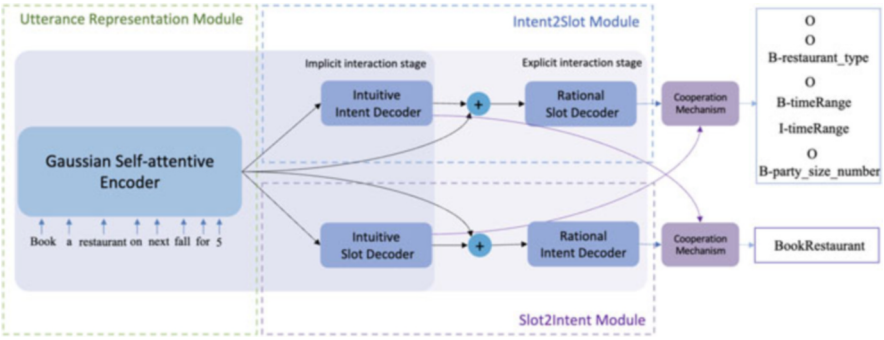

Fig. 3.12 Illustration of a Parallel Interactive Network (PIN) for joint intent detection and slot filling [278]

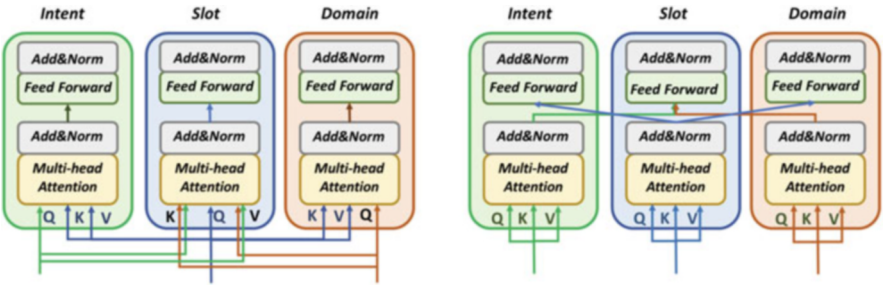

Fig. 3.13 Architecture of the Cross-Attention (left) and before-feed-forward (right) Tri-Transformer-Encoder variants [231]

Further, the concept of what context the transformer is capturing was considered. In its now standard setup of global self-attention, the thought is that local context details important to slot filling may lose out to global detail important to intent. To redress this, [278] used a Gaussian transformer, which aims to better capture local context (Fig. 3.12). Rather than a common encoder which better suits global or local context, the more recent pattern is a move to separate specialized encoders. Qin et al. [165] used independent transformer encoders for slot and intent representations, and [88] used a BiLSTM (suiting local context) for slots and a transformer (suiting global context) for intent. Weld et al. [231] used separate BERT encoders for the inputs to each of the domain, intent, and slot sub-tasks (Fig. 3.13).

3.4.1.4 Hierarchical Models

In a hierarchical model, layers are assigned to the sub-tasks, typically in order of decreasing granularity. Analysis occurs at the base level, and signals of a strength above a threshold are passed to the next layer for scrutiny. For example, in [115], token vector representations are generated in the slot layer, and these are combined in a cell for each intent label. These are then further amalgamated to a domain representation. In this model, the information flow is unidirectional, from slot to intent to domain. Zhang et al. [265] addressed this by providing feedback from the top of the hierarchy to the base. In their capsule network, the layers are word, slot label, and intent label. At each level, a capsule is a cell of neurons whose output can be used for predictions at the next level; word capsule outputs can be used to make slot label predictions, and slot label capsule outputs are used to make intent predictions. A intent message at the highest level, if above a certain threshold, is fed back as an input to the word-level capsules. Staliūnaitė and Iacobacci [200] also used a hierarchical capsule network with extra layers for Named Entity Recognition (NER) and Part-Of-Speech (POS).

3.4.1.5 Bidirectional Models

We have previously discussed unidirectional models, where there is an explicit flow of information from one sub-task to the other. We defined slot2intent and intent2slot variations. The information exchange observed in hierarchical models is explicit; however it is often indirect. For example, in the feedback arrangement of [265], the slot signal directly feeds the intent layer, but the intent signal is fed to the slot layer via the preliminary word layer.

Bidirectional models aim to provide information from each sub-task to the other at some intermediate stage in their respective paths. The term bidirectional here is different from the bidirectional consideration of context used in RNN models. Typically, these networks do not use feedback rather there exist parallel paths through the circuit, each focusing on one of the sub-tasks. There may still be a common encoder level at the start of the circuit and a fusion layer (or a joint loss) at the end.

In [225], each path is a BiLSTM, and the hidden states from each path are shared with the other to form the path output. LSTM decoders on each side to generate the intent and slot label predictions. The loss is not a joint loss, but the circuit performs asynchronous training. In this scheme, the circuit first predicts intent for a batch, and backpropagates intent loss, then it predicts slots for the same batch and backpropagates slot loss. Bhasin et al. [16] simply experimented with different fusion approaches (addition, average, and concatenation) from their separate paths, which were convolutional for intent, and BiLSTM/CRF for slot. Haihong et al. [52] also considers bidirectionality; however, their network can only operate in either slot2intent mode or intent2slot mode at one time (Fig. 3.14).

Explicit, direct bidirectionality with intent2slot and slot2intent paths is used in the Bi-ED circuit of [75]. The intent2slot path combines initial softmax intent prediction on a BERT sentence embedding with the BERT word token embeddings. In parallel, slot2intent combines initial softmax slot predictions on BERT word embeddings with the BERT utterance embedding. The technique was extended to a tri-level semantic frame including domain in [231]. Sun et al. [202] similarly use such preliminary probabilities.

Qin et al. [165] and Hui et al. [88] use separate slot and intent encoders then add sections within following transformer layers to facilitate interaction between the two, a method also explored for the extended semantic frame by [231].

3.4.1.6 Memory Networks

Similarly to hierarchical models, [128] also consider a pipeline from words to slots to intent. Rather than feedback they incorporate a more bidirectional interaction, alternating interaction from slots to intent and vice versa via stacked memory blocks.

A memory block constructs token-level slot features, intent features, and hidden states. These are randomly initialized and updated at each step. Attention takes place globally between the memories in each block, and locally between the elements of

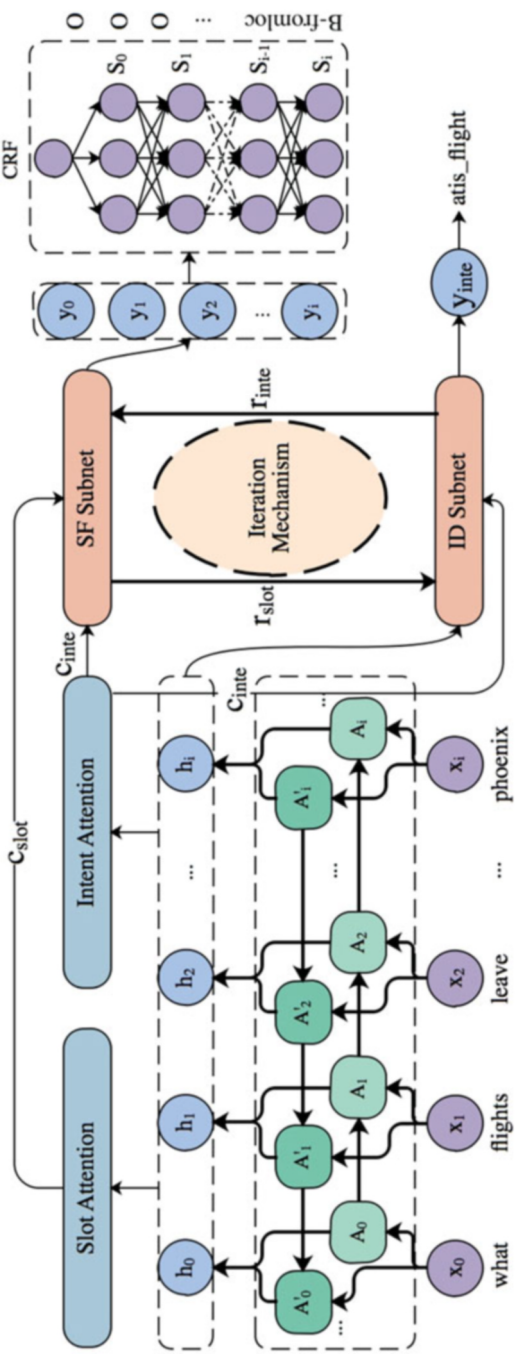

Fig. 3.14 The structure of the proposed model based on SF-ID network [52]

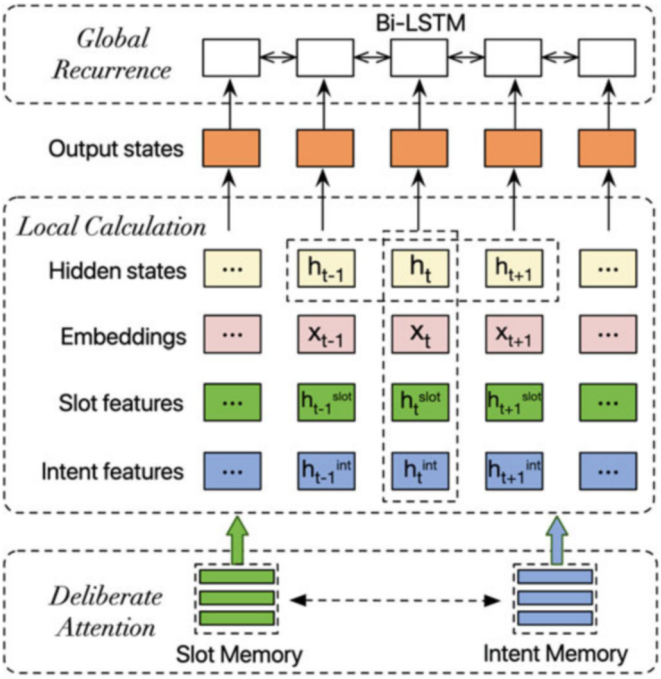

Fig. 3.15 The internal structure of our CM-Block, which is composed of deliberate attention, local calculation and global recurrent respectively [128]

an individual memory block. LSTM elements in each block also address sequential recurrence patterns. After the stacked blocks, a final prediction takes place. Slots are labelled via a CRF using the final hidden states, while intent uses an average of the hidden states in the final block (Fig. 3.15).

3.4.1.7 Multi-task Architectures

A more general study of the optimal flows for the three task problems was undertaken in [161]. For feature creation, they consider a single common encoder versus separate task-specific encoders versus various combinations of two encoders. Then for analysis they consider separate parallel task-specific decoders versus series decoders. No attention or slot gating occur. Other varieties of series architectures from multi-task circuit design are also tested in [59].

3.4.1.8 Graph Networks

RNNs and CRFs learning of global relationships can be lacking as they focus on local context windows. Graph networks are used to address such shortcoming as they can learn global relationships between words and labels.

Zhang et al. [267] construct a graph for each sentence where the nodes are the word and sentence representations from an LSTM. Only word nodes within a context window are connected by edges, while the sentence node is connected to all word nodes. Messages are passed between the joined nodes over multiple time steps, and the node representations are updated. The final word node representations go through a convolution unit and self-attention before informing the slot filling. The final sentence node representation is used for intent detection.

Tao et al. [207] instead proposes a global graph. They construct syntactic dependency trees for the training sentences. The words in the vocabulary form the node set. If words are connected in a dependency tree, they are joined by an edge in the graph incorporating the type of the dependency. A graph convolutional network (GCN) then provides representations of the words and dependency types. A gating mechanism then combines BERT embeddings with a lookup of the GCN for use in the joint task.

3.4.1.9 Importing Methods from Analogous Fields

Bhasin et al. [17] propose that the relationship between intent and slots is similar to that between the query and image in visual question answering. Borrowing from the latter field, they perform Multimodal Low Rank Bilinear (MLB) fusion between the intent and slot features. Ni et al. [151] compare the joint task of clinical domain detection and entity recognition in medical literature. They propose a pipeline structure where intent is detected first and then slots are determined. Namazifar et al. [149] recast the joint task as a Question Answering problem and use an encoder-decoder transformer to ask binary intent questions ("is the intent asking about ____?") and information gathering slot questions ("what cuisine is mentioned?"). The use of LLMs for NLU can be seen as another way of recasting the joint task as Question Answering.

3.5 Single-Turn NLU with Datasets

Datasets for Natural Language Understanding (NLU) include natural language utterances that are annotated at the sentence level for intent, at the token level for slot labels, or both. These datasets can come from a single domain, such as ATIS or FRAMES, or from multiple domains, like SNIPS-NLU or MTOP.

One source of these datasets is spoken language transcribed using automatic speech recognition software. Examples include requests to personal electronic

assistants (SNIPS-NLU, Cortana, Alexa) and conversation logs with human agents (MultiWoz, ATIS, DSTC sets). In the case of M2M (Machines Talking to Machines), a generative model mimics human conversations. Another source is typed sentences, such as those found in CONDA from in-game chat or CQUD, which uses typed sentences from a Chinese question-answering Web site. Typed sentences can include emoticons, providing clues about the speaker's intent. These utterances may be processed for spelling, case, and punctuation or left unchanged for researchers to explore different pre-processing techniques.

Single-turn NLU datasets contain isolated utterances, with each sentence standing alone. In contrast, multi-turn datasets include sequences of related utterances, typically between two speakers. Multi-turn datasets allow the use of previous utterances as context to analyze the current sentence. Single-turn datasets may cover multiple domains but are often not annotated for each domain, a limitation that researchers or dataset providers could address. Multi-turn, multi-domain datasets, however, often include domain information, adding an extra layer of context about the conversation's subject matter.

Multi-intent datasets can detect more than one intent in an utterance, while multilingual datasets enable experiments on transfer learning and model generalizability across languages.

Table 3.2 provides information on common NLU datasets. ATIS and SNIPS-NLU are popular due to their availability and wide use, allowing for model comparisons. However, there are issues with these benchmarks, which we will describe next. Statistics for these datasets are provided in Table 3.3.

3.5.1 ATIS Dataset

The Air Travel Information System (ATIS) is a single-turn English dataset that captures requests for air travel information, such as flights, fares, airlines, airports, in-flight meals, and ground services. It is the longest-standing NLU dataset, and understanding its development helps us grasp the field's conventions. The first version, ATIS-0, was introduced in 1990 by [80]. It included 740 samples containing elements of an end-to-end dialogue system, including sound files, transcriptions, a flight information database, answer tuples, and the SQL query generating those tuples. Transcription tokens were created according to Standard Normal Orthographic Representation (SNOR) rules: case-insensitive alphabetic text with white-space-separated lexical tokens, letters followed by a full-stop for spelled-out letters (e.g., "a. b. c."), and only apostrophes for contractions and possessives, and hyphens for hyphenated words and fragments. The average length of the SNOR-translated utterances was 11.3 tokens. Subsequent versions, ATIS-1 [157], ATIS-2 [84], and ATIS-3 [44], included sound files and extended databases and were available from late 1993 to mid-1994. The standard ATIS for NLU analysis was based on samples from ATIS-2 and ATIS-3. In 2003, [262] continued designing end-to-end dialogue systems but shifted focus toward NLU. Their work on semantic

Table 3.2 Major datasets used in the literature, single turn in English unless noted, Train-Val-Test gives the number of utterances

Name	Public	Train-Val-Test	Num Intent	Num Slots	Domain, Notes
ATIS	Y	4478/500/893	21	128	Air travel
SNIPS-NLU	Y	13084/700/700	7	72	Personal assist.
FRAMES	Y	20006/-/6598	24	136	Hotel, multiturn
CQUD	N	3286	43	20	Chinese, question answering
TREC	Y	5500/-/500	6(50)	–	Question classification
TRAINS	N	5355/-/1336	12	32	Problem solving, multiturn
Microsoft Cortana	N	10k/1k/15k	10–20	15–63	Personal assist., multidomain
Facebook	Y	30521/4181/8621	12	11	Multi-lingual task oriented
SRTS FrameNet	N	2803/-/312	12	61	Robotics
Alexa	N	264000/-/-	246	3409	17 domains
DSTC2	Y	4790/1579/4485	13	9	Multiturn, restaurant search
DSTC4	Y	5648/1939/3178	87	68	Multiturn, tourism dialogue
DSTC5	Y	27528/3441/3447	84	533	Dialogue with social robots
CMRS	N	2901/969/967	5	11	Chinese, room reservations
CU-Move	N	57584/-/-	5	38	In-vehicle dialogue
AMIE	N	3418/-/-	10	7	In-vehicle dialogue
TeleBank	N	2238/-/-	25	17	Korean, banking
CONDA	Y	26921/8974/8974	4	6	In-game chat
MTOP	Y	73174/10453/20907	117	78	11 domains, 6 languages
MIT Movie_Eng	Y	8798/97/2443	–	25	Movies, slot only
MIT Restaurant	Y	6894/766/1521	–	17	Restaurants, slot only
mixATIS	Y	13162/759/828	18	128	Air travel
mixSNIPS	Y	39776/2198/2199	7	72	Personal assist

parsing laid the groundwork for today's slot labelling. In 2007, [179] used the same dataset but updated the annotations, resulting in the familiar ATIS format with *null* slots matching today's 'O' labels. This version includes 4978 training samples and 893 test samples, with 17 base intents, later expanded to 21 intents through joint intents. It also relaxed some SNOR rules, allowing punctuation, lowercase text, and numbers for times and years (but not dates). In 2010, [213] analyzed ATIS using AdaBoost and CRF methods. They performed an AdaBoost classification on word n-gram features for intent classification and a separate CRF-based method for slot

Table 3.3 ATIS and SNIPS dataset statistics

Dataset	ATIS	SNIPS
Vocab size	722	11,241
Average sentence length	11.28	9.05
#Intents	21	7
#Slots	120	72
#Training samples	4,478	13,084
#Validation samples	500	700
#Test samples	893	700

labelling. They classified errors into six types for intent and five types for slots and suggested research directions based on these errors: use of parsers to identify head words or clauses, use of a priori information, and methods to enable long-distance pattern identification rather than shorter local patterns. Some of these directions have been incorporated into deep learning through the use of knowledge bases and recurrent neural networks. They also noted high misannotation rates (2.5% for intent and 8.4% for slots).

Eight years later, [13] conducted another analysis to evaluate the usefulness of ATIS. They tested various methods, from a boosted tree ensemble to a BiLSTM net, on ATIS slot tagging with and without named entity tag labelling. They used the same data as [179], addressing issues with BIO position labels by treating semantic spans as single tokens. For example, "san jose" is treated as one token, not two. Although this approach has weaknesses, the results are valuable. They compared errors from their five best models, grouping predicted slots into:

- agree/correct (AC) - all models correctly identify the slot and agree on the answer.
- non-agreement/error (NE) - all models incorrectly identify the slot and disagree on the errors.
- agree/error (AE) - all models incorrectly identify the slot and make the same error.
- non-agreement/correct (NC) - models don't agree on the solution, but at least one is correct.

These clusters point to future research directions. While the "agree/correct" (AC) cluster is "solved," the "agree/error" (AE) and "non-agreement/error" (NE) clusters remain open issues, indicating aspects of slot labelling not captured by current models. The "non-agreement/correct" (NC) cluster is useful for comparing models that successfully identified the slots with those that did not.

Further issues with the dataset's annotations were identified and classified. For example, there is ambiguity where a value could fit different labels, and repetition errors where a token is repeated but only labelled once. They estimated that approximately 2.5% of the utterances were incorrectly slot-tagged, suggesting that ATIS might be nearing the end of its usefulness for analysis. The next comprehensive

analysis of the ATIS dataset was conducted by [153], who thoroughly reviewed its shortcomings. They re-annotated the dataset to correct what they identified as errors and recommended future work to consider using this revised version. Even without the re-annotated version, literature results indicate that test intent accuracy is now above 99% and slot F1 above 98%. Current models appear to have effectively captured the joint distributions of words, slots, and intents in the dataset, including the error rates reported by [213]. Future models may only yield minor improvements, which might not significantly enhance evaluation metrics. In the current version of the dataset, the intents are highly imbalanced, with 75% of the samples in the "flight" intent. This imbalance can be addressed using machine learning balancing techniques, though most models perform well without them. This imbalance also sets ATIS apart from the next benchmark dataset.

3.5.2 SNIPS Dataset

The SNIPS Natural Language Understanding (SNIPS-NLU) dataset is comprehensively described in [41]. This dataset consists of 15884 utterances, categorized into 7 balanced intent classes. For training purposes, it includes 72 slot labels and a vocabulary of 11241 words, with an average sequence length of 9.05 words. Unlike ATIS, which is limited to a single domain, SNIPS covers three domains: weather, restaurants, and entertainment. A visualization by [128] demonstrates that the slot labels for different domains are mostly distinct (Fig. 3.16). These differences make SNIPS a valuable counterpart to ATIS for research in Natural Language Understanding (NLU). Despite these distinctions, SNIPS achieves very high test results, with intent accuracy exceeding 99% and slot F1 scores around 98%.

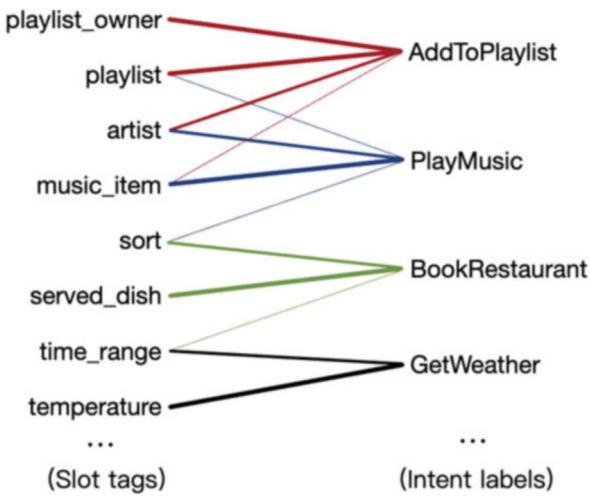

Fig. 3.16 Statistical association of slot tags (on the left) and intent labels (on the right) in the SNIPS, where colors indicate different intents and thicknesses of lines indicate proportions [128]

3.5.3 Other Datasets

Microsoft has several proprietary datasets used by their researchers. For example, FRAMES contains multi-turn dialogues related to hotel bookings. The Microsoft Cortana personal voice assistant datasets encompass at least six domains: weather, calendar, communication, reminders, alarms, and places.

Certain competitions have also provided valuable datasets for multi-turn dialogues. These include DSTC 2, 3, and 5, as well as the Chinese competition-based CCKS dataset. The TRAINS dataset features problem-solving dialogues. Other notable datasets come from various fields, such as FrameNet from robotics, CU-Move and AMIE from in-vehicle communication, CONDA [233] from in-game dialogue, and CQUD (from Baidu Knows), Yahoo, and TREC (only intent annotated) from question answering.

There are also datasets in languages other than English. Bellomaria et al. [14] developed an Italian dataset by translating SNIPS and using Italian words for tokens like city and movie names. ATIS has been translated into multiple languages, resulting in MultiATIS++ [247]. For multilingual transfer learning, [186] provided a tri-lingual dataset, while MTOP offers nearly parallel utterances across 11 domains and 6 languages [119].

The mixATIS and mixSNIPS datasets [166] combine sentences from ATIS and SNIPS to create multi-intent sentences, with ratios of 0.3:0.5:0.2 for sentences with 1, 2, and 3 intents, respectively.

3.5.4 Discussion

ATIS and SNIPS are approaching the end of their usefulness as benchmarks for joint tasks. They face challenges such as mis-annotation, out-of-vocabulary (OOV) issues, and potentially unnatural language. Despite these problems, very high test results show that methods developed in the field can successfully learn the joint distributions of intent and slot labels and the relationships between slot labels in a supervised learning setting.

These datasets are still valuable for study because they contain single utterances, have reasonable numbers of intents and slots, are task-focused (with clear intent), and have suitable utterance lengths. They also offer important differences, such as class imbalance versus balance and single versus multi-domain. A model that performs well on both datasets can claim to have some generalized capability. However, the literature notes that the generalizability of such supervised learning models to new domains remains uncertain.

These datasets are likely to remain benchmarks because they allow comparisons of new approaches to previous ones.

It is essential to test more naturally conversational datasets. Examples include DSTC, M2M, and MultiWOZ.

To avoid the high cost of annotation, new datasets could be largely unannotated, encouraging research in zero-shot or few-shot methods. Metrics for evaluating the effectiveness of such models without annotation need to be considered. Notably, all the few-shot and zero-shot papers reviewed use annotated datasets for evaluation, such as testing on ATIS and SNIPS [109].

Generative models (LLMs) might be used to annotate datasets. However, this technology is still in its early stages, and their performance, especially in slot labelling, is inferior to the specialized models discussed in this book.

Chapter 4
Multi-turn Natural Language Understanding

4.1 Overview

As we journey deeper into Natural Language Understanding (NLU), we now turn our attention to multi-turn NLU, a sophisticated frontier where dialogue systems engage in extended interactions with users, weaving a tapestry of context and continuity. Multi-turn NLU is the cornerstone of advanced conversational agents, enabling them to remember, reference, and build upon previous exchanges, thus providing a seamless and coherent conversational experience.

Figure 4.1 presents a multi-turn conversation example that shows the annotation method for word-level slots, sentence-level intent, and conversation-level domain from the M2M [188] dataset. The dialogue consists of a total of nine turns, each including word-level slot tokens and sentence-level intent information, with the dialogue corresponding to one domain, 'restaurant'.

This chapter unfolds the trends and complexity of multi-turn NLU, beginning with exploring NLU with domain-specific techniques. Here, we delve into the methods tailored to understand and manage dialogues within specific domains. These techniques employ context-aware strategies, leveraging the nuances and particularities of each domain to enhance comprehension and relevance in conversations. From dialogue state tracking to context propagation, we examine the arsenal of techniques that empower systems to maintain and utilize context across multiple turns.

Following this, we navigate through the landscape of external resources, which serve as vital augmentations to multi-turn NLU systems. These resources include vast knowledge bases and dynamic knowledge graphs that infuse conversations with depth and breadth of information. By integrating these external sources, dialogue systems can draw upon a rich repository of facts, relationships, and insights, thereby enriching the dialogue with contextually appropriate and timely information. This section also covers the innovative use of external memory networks and the incorporation of domain-specific lexicons and ontologies.

S. C. Han et al., *Natural Language Understanding in Conversational AI with Deep Learning*, https://doi.org/10.1007/978-3-031-74364-1_4

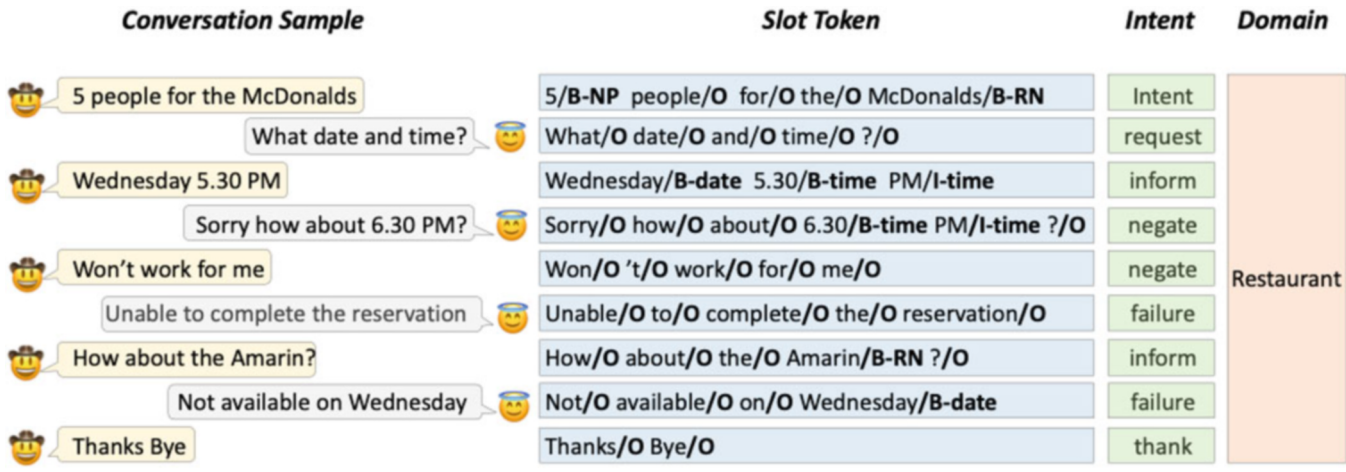

Fig. 4.1 An example of conversations with word-level slots, sentence-level intent, and conversation-level domain annotation from **M2M**

To anchor our discussion in practical applications, we then explore the datasets that form the bedrock of multi-turn NLU research and development. These datasets span various domains and interaction types, providing invaluable benchmarks for training and evaluating conversational agents. Among these are prominent datasets like **MultiWOZ** [22], which offers a rich collection of dialogues across multiple domains, and **DSTC** [237] series datasets, renowned for their focus on dialogue state tracking and multi-turn interactions. We also highlight datasets designed for specific applications, such as in-vehicle dialogue, customer service, and more, showcasing the diversity and complexity of real-world conversations.

By the end of this chapter, readers will have gained a comprehensive understanding of the techniques, resources, and datasets pivotal to mastering multi-turn NLU. This knowledge will empower them to develop and refine dialogue systems that not only understand and respond to user inputs accurately but also engage in meaningful, coherent, and contextually aware conversations that span multiple turns, mirroring the fluidity and richness of human dialogue.

4.2 Multi-turn NLU with Datasets

In our quest to unravel the secrets of multi-turn Natural Language Understanding (NLU), we find ourselves at a critical juncture: the dataset! These repositories of human conversation are the lifeblood of any dialogue system and provide the raw material from which our models learn to comprehend and respond to multi-turn interactions.

In multi-turn Natural Language Understanding (NLU), datasets play a crucial role in training, evaluating, and refining models. These datasets provide the necessary context and examples for models to learn how to handle multi-turn dialogues effectively. This sub-chapter delves into the key datasets used in multi-turn NLU and explores the advanced techniques employed to leverage these datasets to create sophisticated conversational agents.

4.2.1 Characteristics of Multi-turn NLU Datasets

Multi-turn NLU datasets are characterized by the depth and complexity of multi-actor conversation. Unlike single-turn datasets, where each interaction is an isolated event, multi-turn datasets weave a continuous dialogue thread, capturing the ebb and flow of conversation. They contain multiple layers of information.

Imagine you are in a bustling restaurant, alive with the clatter of cutlery and the murmur of conversation. This is not merely a setting but a dynamic environment where context is king. Each exchange between patrons and staff is a turn in the grand dining dialogue, rich with intent, conversation context, and subtle word-token

cues. It is within such nuanced interactions that the true challenge of multi-turn NLU lies.

- **Word-Level Slots:** These are the fine-grained threads of the conversational tapestry. Each word may carry specific semantic information crucial for understanding the intent behind an utterance. For instance, in our bustling restaurant, words like "table," "reservation," and "vegetarian" might be tagged to capture their specific roles.
- **Sentence-level Intents:** Each sentence, like a brushstroke in a painting, contributes to the overall picture of the conversation. Sentence-level intents encapsulate the purpose of each utterance—be it making a reservation, ordering food, or enquiring about dietary options.
- **Conversation-Level Domains:** The most important theme of the dialogue, the domain, provides the context that binds the conversation together. In our restaurant scenario, the domain might be "dining experience," guiding the interpretation of each turn within that context.

To illustrate these principles, let us explore some of the most renowned datasets in the field of multi-turn NLU:

4.2.1.1 The Dialog State Tracking Challenge (DSTC)

The Dialog State Tracking Challenge (DSTC) [237] series offers a comprehensive suite of datasets meticulously designed for benchmarking dialogue state tracking systems, which are crucial for multi-turn NLU. These datasets encompass a wide variety of dialogue scenarios, each labelled with annotations that capture the dialogue state, user intents, and system actions. This detailed labelling provides a rich data source for training models to maintain and update the state of a dialogue accurately, ensuring that the system can interpret and respond to user inputs effectively over multiple turns. The versions of DSTC that prominently feature multi-turn dialogues include:

- **DSTC2** [81] and **DSTC3** [82]: Focused on single-domain dialogue state tracking but involved multi-turn interactions where systems had to maintain context across turns within a given domain (e.g., restaurant reservations).
- **DSTC4** [102]: Introduced human-human dialogues, adding complexity to the state-tracking tasks and emphasizing multi-turn interactions.
- **DSTC5** [103]: Continued the focus on human-human dialogues, particularly in travel planning, which involves extended multi-turn interactions.
- **DSTC6** [86]: Included tasks that required systems to handle multi-turn dialogues in a more naturalistic setting, such as negotiation dialogues.
- **DSTC7** [49]: Featured tasks like End-to-End Conversation Modelling and Noetic End-to-End Response Selection, which involve multi-turn interactions to a significant extent.

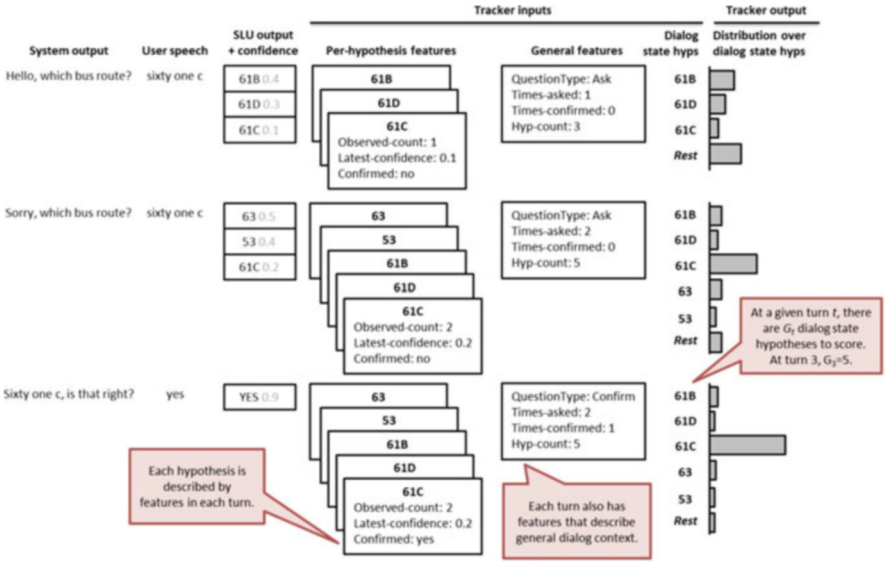

Fig. 4.2 Overview of dialog state tracking. In this example, the dialog state contains the user's desired bus route [237]

- **DSTC8** [104]: Introduced a new task specifically focused on multi-domain dialogue state tracking, requiring systems to handle multi-turn dialogues across different domains.
- **DSTC9** [70] and **DSTC10** [256]: These latest iterations continue to push the boundaries with tasks involving multi-turn and multi-domain dialogue state tracking, along with other advanced dialogue challenges.

Those DSTC series have become a cornerstone in the field, extensively utilized by researchers and practitioners alike for both training and evaluating multi-turn dialogue state tracking models. Its datasets enable the development of systems that are capable of understanding and managing complex conversational dynamics, thereby advancing the capabilities of conversational AI (Fig. 4.2).

4.2.1.2 MultiWOZ (Multi-domain Wizard-of-Oz)

MultiWOZ (Multi-domain Wizard-of-Oz) [22], including **MultiWOZ 2.1** [54] and **MultiWOZ 2.2** [263], is designed to simulate real-world conversations involving multiple domains. It includes dialogues where users interact with a system to accomplish tasks such as booking hotels, reserving restaurants, finding attractions, and more. While MultiWOZ 2.1 focused on correcting major annotation errors and improving consistency, MultiWOZ 2.2 went further by making additional

Fig. 4.3 Example of the difference between dialogue state annotation in MultiWOZ 2.1 and MultiWOZ 2.2 and span annotations in MultiWOZ 2.2 [263]

corrections, aligning the dataset schema with task requirements, and enhancing overall data quality and consistency (Fig. 4.3).

The dataset overall covers multiple domains, allowing for the development of systems that handle diverse conversational contexts. Each dialogue is annotated with user intents, system actions, and dialogue states, providing a rich source of labelled data for training models. The dialogues often involve complex interactions with context switches and dependencies, making it ideal for testing the capabilities of multi-turn NLU systems.

MultiWOZ has become a benchmark dataset in the field of NLU, widely used for various research purposes:

1. **Dialogue State Tracking:** Researchers use MultiWOZ to train and evaluate models for tracking the state of the conversation, maintaining context across multiple turns.
2. **Intent Recognition and Slot Filling:** The dataset is instrumental in developing systems that can accurately identify user intents and extract relevant information (slots) from user inputs.
3. **Context Management:** MultiWOZ facilitates studying how dialogue systems manage context and continuity in multi-turn interactions, leading to advancements in context-aware NLU techniques.

4.2.1.3 CamRest676

CamRest676 [235, 236] is one of the widely used multi-turn conversational datasets specifically tailored for restaurant reservations. Collected and curated to advance

dialogue state tracking, it offers a wealth of multi-turn interactions that reflect real-world complexities and contextual dependencies. It comprises approximately 676 dialogues, each centered around booking a restaurant in Cambridge. These dialogues include a variety of conversational scenarios, ranging from simple inquiries about restaurant availability to more intricate interactions involving specific preferences and requirements. The dataset is designed to simulate the natural flow of multi-turn human conversation, making it a valuable resource for training and evaluating dialogue systems.

The multi-turn nature of CamRest676 is one of its representative features. Each dialogue consists of multiple turns, with users and systems engaging in back-and-forth exchanges that require the system to maintain and utilize context effectively. This continuous interaction is critical for several reasons:

- **Context Retention:** Systems must remember previous exchanges to provide consistent and relevant responses. For example, if a user mentions their preference for vegetarian options early in the conversation, the system must retain this information and suggest suitable restaurants in subsequent turns.
- **Multi-turn Intent Understanding:** Each turn may introduce new intents or refine existing ones. A user might start by asking for a restaurant recommendation and later specify a preference for a particular cuisine or price range. The system must track and update these intents throughout the dialogue.
- **Slot Filling:** In multi-turn conversations, slot values (e.g., location, time, cuisine type) are often provided incrementally. The system needs to collect and integrate these values over multiple turns to successfully fulfill the user's request.
- **Multi-turn Dialogue Management:** Their dialogue management involves handling interruptions, clarifications, and changes in user preferences. CamRest676 includes such scenarios, challenging systems to adapt and respond dynamically.

Annotation Schema: CamRest676 is annotated with word-level slots and sentence-level intents. This detailed annotation ensures that the dataset captures the complexity of each conversation. The annotations include:

- **Informable Slots:** These are the information users provide to the system, such as desired area, cuisine, and price range.
- **Requestable Slots:** These are the details users might request from the system, such as address, phone number, and restaurant rating.
- **Dialogue Acts:** Each turn is tagged with dialogue acts like "inform," "request," "confirm," and "deny," which specify the function of the utterance within the conversation.

Engaging with the CamRest676 dataset reveals a deeper appreciation for the complexities and nuances of human dialogue. It stands as a testament to the potential of multi-turn NLU to transform how we interact with machines, paving the way for more natural and intuitive conversational agents.

4.2.1.4 ConvAI2 (Conversational Intelligence Challenge)

ConvAI2 [50] is a dataset originating from the Conversational Intelligence Challenge, specifically focused on open-domain conversations. This dataset comprises dialogues derived from human-bot interactions, enriched with annotations that detail dialogue acts and user feedback. Such annotations provide invaluable insights into the dynamics of open-domain interactions, enabling researchers to train chatbots and conversational agents more effectively. By utilizing ConvAI2, developers can enhance their models' ability to manage and engage in diverse and unrestricted conversational contexts.

ConvAI2, derived from the Persona-Chat [269] dataset, comprises thousands of dialogues where participants engage in conversations by embodying personas' predefined character profiles with specific traits and preferences. This persona-driven approach introduces a unique layer of complexity, as systems must not only maintain context across multiple turns but also consistently reflect the persona's attributes throughout the conversation.

The multi-turn nature of the ConvAI2 dataset is integral to its design. Each dialogue involves an extended series of exchanges, requiring the system to manage and leverage context dynamically. The dataset's multi-turn characteristics are crucial for several reasons:

- **Contextual Continuity:** Systems need to track and reference previous turns to ensure coherent and contextually appropriate responses. This continuity is essential for maintaining a natural flow in the conversation, particularly when discussing topics that span multiple turns.
- **Persona Consistency:** Each participant's persona includes details such as hobbies, interests, and personal preferences. The system must integrate these details into its responses, ensuring the dialogue remains consistent with the persona's profile across all turns.
- **Dynamic Dialogue Management:** Effective management of multi-turn conversations involves handling various conversational elements such as clarifications, topic shifts, and elaborations. The ConvAI2 dataset includes systems that adapt and respond consistently and dynamically.

Annotation Schema: ConvAI2 is annotated with multi turn-level dialogue acts and persona-related information. The annotations are designed to capture the subtleties of each exchange, providing a robust framework for training and evaluating dialogue systems.

- **Persona Sentences:** Each participant's persona is represented by a set of sentences describing their attributes and interests. These sentences guide the dialogue.
- **Dialogue Acts:** Each turn of conversation is tagged with dialogue acts like "statement," "question," and "response," indicating the function of the utterance within the conversation.

- **Contextual References:** Annotations highlight references to previous turns, ensuring systems maintain contextual and persona continuity throughout the dialogue.

Created for the Conversational Intelligence Challenge 2 (ConvAI2), this dataset captures the essence of multi-turn interactions, showcasing the depth and complexity of natural dialogues.

4.2.1.5 Machines Talking to Machines (M2M) Dataset

M2M [188] is a cutting-edge resource designed to push the boundaries of multi-turn conversational AI. Unlike traditional datasets, M2M focuses on dialogues between artificial agents, simulating complex and multi-turn interactions.

M2M is crafted to facilitate research in dialogue systems and aims to achieve coherent, contextually aware conversations across multiple turns. This dataset is particularly valuable for its novel approach to capturing the complexity of machine-to-machine communication, providing a robust framework for training and evaluating advanced conversational agents.

- **Contextual Continuity:** Systems must maintain context across turns. This involves tracking entities, intents, and dialogue history to provide a consistent machine-to-machine conversational flow.
- **Dynamic Dialogue Management:** Multi-turn conversations in M2M are designed to simulate real-world scenarios by machines where topics can shift and evolve, requiring dynamic management. Systems handle interruptions, clarifications, and changes in the conversational trajectory.
- **Task-Oriented Interactions:** Many dialogues in M2M are task-oriented, requiring systems to achieve specific goals through extended interactions. This involves understanding and executing multi-step instructions, managing dependencies, and ensuring successful task completion.
- **Richer Annotations:** Each dialogue is annotated with detailed information at various levels, including word-level slots, sentence-level intents, and dialogue-level goals. These annotations provide a comprehensive understanding of each turn and its role in the overall conversation, compared to other datasets.

4.2.2 Creating and Annotating Multi-turn Datasets

Creating a multi-turn dataset begins with the art of data collection. The data is gathered from various sources, capturing the richness and diversity of human interactions. This often involves crowd-sourcing platforms where participants engage in simulated dialogues, mimicking real-world scenarios [163, 169, 193, 220].

Once collected, these raw dialogues are like unrefined gems, brimming with potential but needing careful polishing. Enter the annotators, the master craftsmen

of this process. Armed with expertise and keen attention to detail, they embark on the task of tagging each dialogue turn with layers of meaningful information.

The Art of Annotation Annotation in multi-turn datasets is similar to weaving a complex tapestry. Each thread represents information meticulously placed to form a coherent whole using word-level slots, sentence-level intents, and conversation-level domains.

The data creation process does not end with annotation. The next phase involves refining and validating the dataset to ensure its quality and reliability. Annotated dialogues undergo rigorous checks to correct any inconsistencies and to ensure that the annotations accurately reflect the underlying conversations. Imagine a jeweller inspecting a gemstone, looking for flaws and ensuring its brilliance. Similarly, validation steps ensure the dataset is polished and ready to serve as a robust foundation for training dialogue systems.

The Final Touch Imagine a master chef presenting a beautifully crafted dish, each element perfectly balanced and harmonized. Similarly, the final evaluation of a multi-turn dialogue system involves synthesizing these metrics to provide a comprehensive assessment of its performance. The detailed information can be found in Chap. 5.

In this chapter, we have journeyed through the creation and annotation of multi-turn datasets. These processes are the bedrock upon which advanced dialogue systems are built, enabling them to engage in meaningful, multi-turn conversations that reflect the complexity and vibrancy of real-world interactions.

4.3 Multi-turn NLU Techniques

In the enchanting world of multi-turn Natural Language Understanding (NLU), where dialogues unfold like complicated dance routines, the techniques employed to decipher these conversations are nothing short of magical.

4.3.1 Rule-Based Multi-turn NLU

Our journey begins with the classic elegance of rule-based approaches, similar to a carefully choreographed waltz. In these systems, **predefined rules guide the flow of conversation** [97, 140, 210]. Imagine a dance instructor leading a novice dancer, each step meticulously planned and executed. Rule-based systems operate on similar principles, using handcrafted rules and scripts to manage dialogues.

Each potential user input is anticipated and paired with a corresponding system response, creating a seamless flow of interaction. These systems excel in well-

defined domains where the rules are clear and predictable, ensuring a smooth multi-turn conversational experience. In the context of multi-turn NLU, this involves five key steps for creating rules [23, 139, 140]:

1. **Defining the Domain:** The first step is to clearly define the domain of the dialogue system. For instance, if the system is intended for restaurant reservations, the domain would encompass all possible interactions related to booking a table, menu inquiries, and restaurant policies.
2. **Identifying Interaction Scenarios:** Next, developers identify common interaction scenarios within the domain. These scenarios form the backbone of the rule-based system and outline how users might engage with it. For example, scenarios might include making reservations, asking about opening hours, or requesting dietary information.
3. **Developing Rules and Scripts:** With the scenarios in mind, developers craft specific rules and scripts for each interaction. These rules dictate how the system should respond to user inputs, ensuring that each turn in the conversation follows a logical and coherent path. For instance, a rule might specify that when a user asks for the opening hours, the system should respond with the appropriate information.
4. **Implementing Multi-turn Context Management:** Effective multi-turn interactions require the system to maintain context across multiple exchanges. This involves tracking user inputs and system responses to ensure continuity. For example, if a user mentions a preference for vegetarian options, the system must remember this preference in subsequent turns.
5. **Handling Exceptions and Edge Cases:** No matter how meticulously planned, real-world conversations often introduce unexpected elements. Rule-based systems must be equipped to handle these exceptions and edge cases gracefully. This involves designing fallback rules and responses for unanticipated inputs, ensuring the conversation remains fluid and coherent.

The beauty of rule-based multi-turn NLU systems lies in their predictability and reliability within well-defined domains. Like a perfectly executed dance routine, these systems can deliver a flawless performance when the steps are followed correctly. For instance, in a restaurant booking system, rules can dictate responses to common requests like "book a table for two" or "what are your opening hours?" This predictability makes rule-based systems ideal for applications where interactions are straightforward and repetitive.

However, much like a dancer who struggles when the music changes unexpectedly, rule-based systems can falter in the face of unpredicted conversational twists and turns. Their reliance on predefined rules makes them less adaptable to the dynamic and unpredictable nature of human dialogue. When confronted with inputs that fall outside their scripted scenarios, these systems may produce awkward or irrelevant responses, breaking the flow of conversation.

4.3.1.1 Enhancing Rule-Based Multi-turn NLU Systems

To mitigate these limitations, developers can enhance rule-based systems with three main strategies:

- **Incorporating Flexibility**: Introducing more flexible rules that can handle a broader range of inputs [9, 249] can improve system robustness. This might involve using pattern matching and regular expressions to capture variations in user inputs.
- **Integrating External Resources**: Leveraging external databases and knowledge bases can enrich the system's responses [7, 229], providing more accurate and contextually relevant information. For instance, integrating a restaurant database can allow the system to provide real-time information on availability and menu options.
- **Hybrid Approaches**: Combining rule-based methods with machine learning techniques can create more adaptable [4, 135, 141, 171] and resilient multi-turn NLU systems. For example, machine learning models can handle more complex and variable interactions, while rule-based scripts manage predictable and routine exchanges.

In conclusion, rule-based multi-turn NLU systems represent the elegance of structured dialogue management, offering reliable performance in well-defined domains. Like a carefully choreographed dance, these systems excel when the steps are clear, and the path is predictable. However, to navigate the ever-changing rhythms of human conversation, developers must continually refine and enhance these systems, blending precision with adaptability to create truly engaging and effective dialogue experiences.

In this section, we have explored the foundational techniques of rule-based multi-turn NLU, laying the groundwork. As we continue our journey, we will delve into the dynamic world of machine learning and deep learning techniques, where the dance of dialogue takes on new forms and possibilities, pushing the boundaries of what is possible in natural language understanding.

4.3.2 Machine Learning-Based Multi-turn NLU

Next, we move to the dynamic rhythm of machine learning approaches, where systems learn from data, much like a dancer learns new moves through practice and experience. Traditional machine learning models, including decision trees, support vector machines, and logistic regression, have paved the way for more flexible and adaptive multi-turn dialogue systems.

Machine learning models thrive on annotated datasets, learning patterns and making predictions based on the data they've ingested. In the context of multi-turn NLU, these models can identify user intents and slot values, adapting to the flow

of conversation with greater agility than rule-based systems. Imagine a dancer who, having mastered the basics, can now improvise and adapt to the changing tempo of the music.

Decision Trees [27, 76] Think of decision trees as the fundamental steps in a dancer's repertoire. These models operate by splitting data into branches based on feature values, ultimately leading to a decision or prediction at the leaf nodes. In multi-turn NLU, decision trees can be used to classify user intents or determine the next action in a dialogue sequence.

Support Vector Machines (SVMs) [73, 159] SVMs add a level of precision to the multi-turn understanding dance. They work by finding the optimal boundary (hyperplane) that separates different classes in the data. For instance, in a customer service scenario, an SVM can distinguish between different types of user requests, such as billing inquiries or technical support, based on the features of the dynamic and multi-turn dialogue input text.

Logistic Regression [106, 181, 187] This model brings a smooth and continuous flow, predicting probabilities rather than fixed categories. Logistic regression is particularly useful for binary classification tasks within multi-turn dialogues, such as determining whether a user's request has been fulfilled or if further information is needed.

4.3.3 Adapting to the Domain

Machine learning models excel at adapting to new data, much like a skilled dancer adjusting to the changing tempo of music. In multi-turn NLU, this adaptability is crucial for maintaining context and coherence across multiple exchanges. The machine learning-based multi-turn NLU learning process can be highlighted with three phrases:

Training on Annotated Data Just as dancers practice their moves repetitively, machine learning models are trained on large datasets of annotated dialogues. These annotations include word-level slots, sentence-level intents, and conversation-level goals, providing the multi-turn NLU models with a rich context for learning [22, 121, 131, 188, 235, 236, 248].

Feature Extraction Effective multi-turn dialogue understanding requires identifying relevant features from the input data. In a conversation, these features might include keywords, phrases, syntactic structures, and even contextual cues from previous turns [62, 133, 152, 274]. Feature extraction is similar to a dancer recognizing key beats and rhythms in the music, guiding their movements.

Continuous Learning The dance of dialogue is ever-evolving, with new patterns and scenarios emerging. Machine learning models can be updated with new data, allowing them to learn and adapt continuously. This continuous learning process

ensures that the models remain effective and relevant to the domain and corpus, like a dancer who constantly refines and expands their repertoire [127, 134, 196]

4.3.4 The Power of Ensemble Methods

In machine learning, ensemble methods combine the strengths of multiple models to create a more robust and accurate system. This approach is like a dance troupe, where each dancer brings their unique skills to the performance, creating a harmonious and powerful ensemble.

4.3.4.1 Random Forest-Based Approach

At its core, a **Random Forest** [19] is an ensemble of decision trees, each trained on different subsets of the data. The process begins with creating multiple decision trees, where each tree is constructed using a random sample of the multi-turn dialogue training data [168, 197, 198]. These tasks are crucial for maintaining the flow and coherence of dialogues, ensuring that the system understands user inputs and responds appropriately across multiple turns.

Intent Recognition Random Forests enhance intent recognition by considering various types of features derived from the user's input, such as keywords, syntactic structures, and contextual cues from previous turns [136, 168, 197, 198]. By training on diverse decision trees, Random Forests can capture a wide range of patterns and variations in user intents, leading to more accurate predictions. For instance, when a user says, "I need to book a table for dinner," "I would like to go to italian restaurant," the Random Forest model can effectively recognize the multi-turn intent to make a reservation, even if the phrasing varies.

Slot Filling In multi-turn dialogues, this task becomes more complex as the system needs to maintain context and track slot values across multiple exchanges. Random Forests excel in this task by aggregating the predictions of multiple decision trees, each trained to identify different slot types and values [90, 114]. This ensemble approach ensures that the system can accurately fill slots, even when the information is provided incrementally or with variations. For example, in a dialogue about booking a restaurant, the system can correctly extract and remember details such as the date, time, and number of guests provided across different turns.

Handling Ambiguities and Variations Real-world multi-turn dialogues often involve ambiguities and variations in language. Users might provide information in different orders, use synonyms, or change their requests mid-conversation. Random Forests, with their diverse set of decision trees, are well-equipped to handle these variations. The ensemble nature of the model allows it to generalize better from

the training data, making it more resilient to unexpected inputs and changes in the conversation flow.

4.3.4.2 Gradient Boosting Machines in Multi-turn NLU

In multi-turn NLU, **Gradient Boosting Machines (GBMs)** stand out as virtuoso performers. These models, through a process of sequential learning and error correction, craft a precise predictive model adept at handling the complexities of extended dialogues. Much like a dancer perfecting their routine through iterative practice, GBMs refine their predictions step by step, enhancing the accuracy and fluidity of dialogue systems.

At the heart of GBMs lies the principle of boosting, a technique that builds an ensemble of weak learners—typically decision trees—sequentially. Each new model in the sequence is trained to correct the errors made by its predecessors [61], creating a powerful composite model.

Intent Recognition GBMs excel in this multi-turn dialogue understanding task by leveraging their sequential learning process to refine intent predictions [18, 69, 110]. Each iteration of the model hones in on the intricacies of user inputs, considering contextual cues from previous turns. This iterative refinement results in highly accurate intent recognition, enabling the system to respond appropriately to user queries. For instance, if a user initially asks, "Can you book a table for tonight?" and later provides additional details, the GBM can accurately track and update the intent as the conversation evolves.

Slot Filling Slot filling involves extracting specific pieces of information from user inputs, such as dates, times, and locations. In multi-turn dialogues, slot values may be provided incrementally or across multiple exchanges. GBMs handle this complexity by continuously refining their slot predictions with each new turn, ensuring that the extracted information remains accurate and up-to-date [18]. For example, in a booking scenario, the user might mention the time of reservation in one turn and the number of guests in another. The GBM can seamlessly integrate this information, maintaining an accurate dialogue state.

Context Management Effective multi-turn dialogue systems must maintain context across multiple exchanges. GBMs contribute to complex context management by refining predictions based on the cumulative context of previous turns [68, 184]. This ensures that the system can handle context-dependent queries and provide coherent responses. For instance, if a user initially discusses their dietary preferences and later asks for restaurant recommendations, the GBM can incorporate this context to suggest suitable options.

Handling Ambiguities and Variations Users may express the same intent in different ways or change their requests mid-conversation. GBMs, with their iterative learning process, can adapt to these variations and refine their predictions accord-

ingly [5, 68]. This adaptability enhances the system's resilience to unexpected inputs and changes in the conversation flow.

4.3.5 The Grace of Sequence Models

Moving beyond traditional machine learning models, sequence models add grace and fluidity to the dance of multi-turn Natural Language Understanding (NLU). These models, including Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs), are designed to handle sequences of data, making them exceptionally well-suited for dialogue systems. They capture the temporal dependencies and contextual nuances intrinsic to human conversation, ensuring a coherent and logical flow in multi-turn dialogues.

4.3.5.1 Hidden Markov Models (HMMs)-Based Approach

Hidden Markov Models (HMMs) [12] are based on the concept of Markov processes, where the system transitions between states with certain probabilities. In HMMs, these states are hidden, and only the observed outputs are visible. This makes them particularly adept at modelling the underlying patterns in sequential data.

In multi-turn NLU, HMMs are used to model the sequence of user intents and system responses. Each turn in the dialogue can be viewed as an observable state, while the hidden states represent the underlying user intents and dialogue states. By capturing these sequences, HMMs ensure the dialogue flows logically and coherently [36, 177]. For instance, in a customer support dialogue, HMMs can track the progression from an initial inquiry to troubleshooting steps and finally to issue resolution.

Training and Inference: Training an HMM involves estimating the transition and emission probabilities from a labelled dataset of dialogues. This is typically done using algorithms such as the Baum-Welch algorithm [172]. Once trained, the Viterbi algorithm is used for inference, determining the most likely sequence of hidden states (intents) given the observed sequence of user inputs and system responses.

4.3.5.2 Conditional Random Fields (CRFs)-Based Approach

Conditional Random Fields (CRFs) [113] are undirected graphical models that model the conditional probability of a sequence of labels given a sequence of input data. Unlike HMMs, which model joint probabilities, CRFs model conditional probabilities. This allows CRFs to directly capture the dependencies between labels in a sequence, conditioned on the input data. The structure of CRFs typically

involves nodes representing the labels and edges representing the dependencies between them.

CRFs are particularly useful for slot filling in multi-turn dialogues, where the model needs to consider the context of the entire sequence to make accurate predictions. For example, in a dialogue about booking a flight, the user might provide the destination in one turn and the departure date in another. CRFs can effectively label and segment these slots, ensuring that all relevant information is accurately extracted and maintained throughout the multi-turn dialogue.

Both HMMs and CRFs bring distinct advantages to multi-turn NLU, contributing to the overall harmony of a dialogue system. They ensure that context is preserved across multiple exchanges, maintaining the logical flow of conversation and accurately capturing the user's intent.

In multi-turn dialogues, maintaining context is paramount. Sequence models like HMMs and CRFs excel at tracking the flow of conversation, ensuring that each turn is informed by the preceding context. This enables the system to handle follow-up questions and related queries with ease. For instance, if a user initially asks about a restaurant's location and later inquires about parking options, the system can seamlessly integrate this context to provide relevant and coherent responses. By clear tracking and predicting the sequence of dialogue, HMMs and CRFs ensure that interactions with the system feel more like a coherent conversation with a human user.

4.3.6 The Energy of Deep Learning Multi-turn NLU Approaches

Our exploration then leads us to the boundless energy of deep learning approaches, where the complexity of multi-turn interactions is handled with the grace and power of a professional dancer performing a solo. Deep learning models, particularly neural networks, have revolutionized the field of multi-turn NLU, offering unprecedented capabilities for understanding and generating long-length and multi-turn human conversation.

4.3.6.1 Recurrent Neural Networks (RNNs)-Based Approach

Recurrent Neural Networks (RNNs) [53] bring a rhythmic elegance to the complicated ballet of multi-turn Natural Language Understanding (NLU). These models are designed to handle sequences of data, making them exceptionally well-suited for dialogues that unfold over multiple turns. Imagine a dancer gracefully moving through a complex routine, each step informed by the ones that came before. Similarly, RNNs remember previous inputs, allowing them to maintain context and ensure continuity in conversations. However, like a dancer who occasionally forgets

the steps, RNNs can struggle with long-term memory, presenting unique challenges and opportunities for improvement [182, 261].

The hidden state is updated at each time step based on the current input and the previous hidden state. This update is governed by a set of parameters learned during training. The output at each time step is generated based on the hidden state, ensuring that the network's responses are informed by the entire sequence of inputs [201]. The ability to maintain and update the hidden state allows RNNs to handle sequential dependencies effectively. In a multi-turn dialogue, this means that the network can remember what has been said in previous turns, enabling it to provide contextually appropriate responses. For instance, if a user mentions their travel plans in one turn, the RNN can remember this information and use it to inform responses to subsequent questions about accommodation or itinerary.

Despite their capabilities, traditional RNNs face significant challenges when it comes to maintaining information over long sequences. This is primarily due to issues such as vanishing gradients, which can hinder the network's ability to learn and retain long-term dependencies, like multi-turn conversation.

4.3.6.2 Enhancing RNNs for Multi-turn NLU

There have been developed to address the limitations of traditional RNNs, enhancing their ability to handle long-term dependencies in multi-turn NLU.

Gated Mechanisms One of the most effective enhancements to RNNs is the introduction of gated mechanisms, such as **Long Short-Term Memory (LSTM)** [85] networks and **Gated Recurrent Units (GRUs)** [37]. These mechanisms incorporate gates that regulate the flow of information, allowing the network to maintain long-term memory and mitigate issues related to vanishing and exploding gradients.

Attention Mechanisms [154] Attention mechanisms have also been integrated into RNN architectures to improve their ability to handle long sequences. By allowing the network to focus on relevant parts of the input sequence, attention mechanisms enhance the network's ability to capture long-range dependencies and provide contextually appropriate responses in multi-turn dialogues.

Hierarchical Models Hierarchical RNN models, which process sequences at multiple levels of abstraction, have been proposed to better capture the structure and context of dialogues. These models can handle dependencies at different granularities, ensuring that both short-term and long-term conversation context is preserved [107, 132, 144].

In conclusion, Recurrent Neural Networks (RNNs) brought a dynamic energy to the world of multi-turn NLU, enabling dialogue systems to remember and leverage context across turns sequentially. Despite challenges with long-term dependencies, innovations such as LSTMs, GRUs, and attention mechanisms have significantly enhanced their capabilities. These advancements ensure that RNN-based models

can engage in fluid and coherent conversations, much like a dancer who masterfully navigates a complex multi-turn routine.

4.3.7 Transformer Models

Enter the **transformers** [216], the virtuosos of the deep learning stage. Transformers use self-attention mechanisms to weigh the importance of each word in a multi-turn conversation, ensuring that context is maintained across multiple turns. Imagine a dancer who not only remembers the choreography but also anticipates and adapts to the subtle cues of the music and audience. The self-attention mechanism allows transformers to capture dependencies between words regardless of their distance in the text, making them highly effective for modelling long-range context in dialogues. These models, including the famed **BERT (Bidirectional Encoder Representations from Transformers)** [98] and **GPT (Generative Pre-trained Transformer)** [174], excel in capturing the intricacies of language with remarkable precision.

Transformers revolutionized the field of NLU with their innovative architecture, which forgoes the traditional recurrent mechanisms in favour of attention mechanisms. This departure allows them to process sequences more efficiently and capture long-range dependencies of multi-turn conversation.

Self-Attention Mechanism The core innovation in Transformer models is the self-attention mechanism. For each word in the input sequence, the self-attention mechanism calculates a set of attention scores that indicate how much focus should be given to every other word. This allows the model to consider the entire context of the sequence rather than just the immediate preceding words [254, 266]. This mechanism enables the model to weigh the importance of different words in a sequence when making predictions, capturing the full context and nuances of human conversation.

The attention scores are used to create a weighted sum of the value vectors of the words, producing a contextual representation for each word. This allows the model to capture dependencies between words, regardless of their distance in the long-range conversation. For example, in the sentence "The man sat on the mat," "He was very happy," the model can understand that "he" refers to "The man" through the attention mechanism.

Positional Encoding Since transformers do not inherently process data in a sequential manner, they incorporate positional encodings to retain the order of the words. Positional encodings are added to the input embeddings to provide information about the position of each word in the sequence, ensuring that the model can distinguish between words based on their order. A sinusoidal function generates positional encodings, which allows the model to generalize sequences longer than those seen during training, like a long-range multi-turn conversation [63]. The sine

and cosine functions of different frequencies encode the positions, ensuring that each position has a unique encoding.

Multi-head Attention Transformers use multiple attention heads to capture different aspects of the relationships between words. This multi-faceted attention mechanism allows the model to capture a wide range of dependencies and interactions within the input sequence. Multiple attention heads run in parallel, each computing its own attention scores and contextual representations. This enables the model to capture diverse aspects of the long-range relationships between words [199, 206, 271].

4.3.7.1 Pre-trained Models with Transformer

With the above components, transformers, including its pre-trained models: BERT or GPT, have transformed the landscape of multi-turn NLU, offering unparalleled capabilities in understanding and generating human language. The ability of those pre-trained models and its embedding maintain and understand context across multiple turns makes them ideal for real-world dialog systems that require a deep understanding of conversational flow.

BERT (Bidirectional Encoder Representations from Transformers) BERT [98] excels in understanding the context of a sentence by looking at both the left and right sides of a word simultaneously. This bidirectional approach allows BERT to capture the full context of a word based on its surrounding words, making it highly effective for tasks such as intent recognition, slot filling, and conversation domain detection in multi-turn dialogues [4, 208]. For instance, BERT can understand that "book" in "book a table" refers to making a reservation, while "book" in "read a book" refers to a physical object. Once pre-trained, BERT can be fine-tuned on specific datasets for various NLU tasks. For multi-turn dialogues, this fine-tuning allows BERT to excel in intent recognition and slot filling by leveraging its robust contextual embeddings.

GPT (Generative Pre-trained Transformer) GPT [174] is designed for language generation tasks, making it particularly suited for generating coherent and contextually relevant responses in dialogue systems. By pre-training on vast amounts of text and fine-tuning on specific dialogue datasets, GPT can generate natural and engaging responses that maintain context over multiple turns. For example, in a customer service chatbot, GPT can handle follow-up questions and maintain the conversational context, ensuring a smooth and helpful interaction.

One of the most significant advantages of transformers is their ability to handle long-range context. In multi-turn dialogues, maintaining context over several exchanges is crucial for coherence. Transformers achieve this by using self-attention mechanisms to focus on relevant parts of the conversation, regardless of how far apart they are in the sequence. This capability ensures that important details from earlier in the conversation are not lost, enabling the system to provide accurate and contextually appropriate responses. Transformers learn powerful representations of

data that can capture complex relationships and dependencies. These representations are crucial for integrating external knowledge, as they allow the model to understand and leverage new information in a meaningful way. Rule-based systems do not learn representations, and traditional machine learning models often struggle to learn representations as rich and effective as those generated by transformers. Generally, transformer or transformer-based pertained models involves two key strategies in order to achieve multi-turn NLU research problems:

Dialogue State Tracking It is similar to a meticulous conductor keeping track of every instrument in an orchestra, ensuring that each plays its part at the right moment. In the realm of multi-turn NLU, maintaining the state of the dialogue is crucial for coherent interactions. Transformer models can be integrated with state-tracking mechanisms to keep track of the context and ensure that responses are consistent and relevant. With each user input, the dialogue state is updated to reflect the current context of the conversation. This involves recognizing intents, extracting relevant information (slots), and maintaining a record of the conversation history [238].

Rule-based systems, in contrast, struggle with dynamic context management as they rely on predefined rules that cannot adapt to new information in real-time. Traditional machine learning models, while more flexible than rule-based systems, still face limitations in maintaining long-range dependencies as effectively as transformers.

For instance, in a customer service scenario, if a user first asks about product availability and then about delivery options, the model needs to remember the initial query to provide a coherent and informed response. Transformers use self-attention mechanisms to generate representations of the dialogue state. These representations encapsulate the entire context of the conversation, allowing the model to generate responses that are contextually appropriate [94, 170, 211]. The dialogue state acts as a dynamic memory, constantly evolving with each new turn in the conversation.

Integration with External Knowledge Transformers-based models can be integrated with external knowledge sources, such as databases and APIs [29, 122, 203, 274], to enhance their responses, much like how a conductor might incorporate diverse musical influences into a performance. This integration allows the dialogue system to provide more informative responses to multi-turn utterances by leveraging external data [123]. By incorporating data from external sources, the model can generate responses that are not only contextually appropriate but also enriched with additional information. This makes the interaction more valuable for the user, as the model can provide detailed and relevant answers. External knowledge integration ensures that the model's responses are dynamically updated based on the latest available information. This is particularly useful in scenarios where the information is constantly changing, such as stock availability in an online store or flight schedules in a travel booking system.

Particularly, this fine-tuning process ensures that the model can leverage external knowledge effectively to improve its performance. Rule-based systems lack this flexibility, as integrating new information requires manual updates to the rules.

Traditional machine learning models can be retrained with new data, but this process is not as seamless and efficient as fine-tuning transformers.

4.3.8 Joint Deep Learning-Based Multi-turn NLU

In the grand theatre of multi-turn Natural Language Understanding (NLU), joint deep learning models perform with the harmony and precision of a well-rehearsed ensemble. These models seamlessly integrate multiple tasks, such as intent recognition, slot filling, and dialogue state tracking, into a single, unified framework. Joint models handle the intertwined complexities of multi-turn dialogues, ensuring coherence and contextual accuracy throughout the conversation.

Instead of treating tasks like intent recognition, slot filling, and conversation topic detection as separate entities, joint models approach them as interconnected problems. This integration[11, 212, 232, 240] allows the model to leverage shared information between tasks, leading to better overall performance.

- **Unified Architecture** [18, 212]: Joint models often employ a shared encoder-decoder architecture. The encoder processes the input sequence and extracts contextual features, while the decoder generates outputs for multiple tasks simultaneously. This unified approach ensures that the tasks benefit from each other's contextual understanding.
- **Shared Representations** [166, 232]: By sharing representations between tasks, joint models capture dependencies and correlations between intent and slot information. For example, recognizing that a user wants to "book a hotel" can help accurately identify slots like "check-in date" and "location."
- **Contextual Dependencies** [222]: Joint models excel at handling contextual dependencies across multiple turns in a dialogue. They maintain a coherent representation of the dialogue state, which evolves as the conversation progresses. This capability ensures that the system can handle follow-up questions and context shifts seamlessly.

4.3.8.1 Benefits of Joint Models in Multi-turn NLU

Joint models offer several advantages that make them particularly effective for multi-turn NLU (Table 4.1):

Integrated Learning By training on multiple tasks simultaneously, joint models can leverage shared information and dependencies between tasks [18, 212, 222]. This integrated learning approach improves the model's ability to understand and generate contextually relevant responses. Imagine a pianist who, while mastering a new piece, simultaneously practices both the melody and the harmony. Each hand learns from the other, resulting in a more cohesive performance. Similarly, joint models learn the intricate dance between intents and slots, using the insights from

Table 4.1 A Prediction example with a three-turn conversation on slot filling, intent detection, and domain classification results of each model. The green cell represents a result that matches the ground truth, the red cell indicates incorrect results, and the yellow cell indicates partially correct results

Turn	Model	Tokens (Slot)	Intent	Domain
1	Utterance	boat, is, fine	-	-
	GT	B-RN, O, O	affirm	-
	MIDAS (BERT)	B-RN, O, O	affirm	-
	BERT-Only	O, O, O	inform	-
2	Utterance	sure, ,, i, found, this, one, :, the, view	-	-
	GT	O, O, O, O, O, O, O, B-RN, I-RN	offer	-
	MIDAS (BERT)	O, O, O, O, O, O, O, B-RN, I-RN	offer	-
	BERT-Only	O, O, O, O, O, O, O, O, O	offer	-
3	Utterance	how, about, the, ivy, or, boats, ?	-	-
	GT	O, O, B-RN, I-RN, O, B-RN, O	select	restaurant
	MIDAS (BERT)	O, O, B-RN, I-RN, O, B-RN, O	select	restaurant
	BERT-Only	O, O, B-RN, B-RN, O, B-RN, O	select	movie

one task to enhance the other. This synergy ensures that the model can provide richer, more accurate responses informed by the interplay of different conversational elements.

Consistency and Coherence Joint models ensure that the dialogue remains consistent and coherent across multiple turns. By maintaining a shared representation of the dialogue state, the model can handle context shifts and follow-up questions more effectively [166, 232]. Picture a seasoned conductor guiding an orchestra through a complex symphony, ensuring each movement flows seamlessly into the next. In the same way, joint models track the evolving state of the conversation, ensuring that each response aligns with the preceding context. This continuity is crucial for maintaining the user's trust and engagement as the model navigates the twists and turns of a multi-turn dialogue.

Efficiency Joint models ensure that the dialogue remains consistent and coherent across multiple turns. By maintaining a shared representation of the dialogue state, the model can handle context shifts and follow-up questions more effectively. Picture a seasoned conductor guiding an orchestra through a complex symphony, ensuring each movement flows seamlessly into the next. In the same way, joint models track the evolving state of the conversation, ensuring that each response aligns with the preceding context. This continuity is crucial for maintaining the user's trust and engagement as the model seamlessly navigates the twists and turns of a multi-turn dialogue [238].

Improved Performance The synergistic learning in joint models often leads to improved performance across all tasks. For example, accurate intent recognition

can enhance slot filling by providing context about the user's goals, and vice versa. Imagine a virtuoso violinist whose mastery of bowing techniques enhances their fingering precision, resulting in a more expressive and nuanced performance. In joint models, the mastery of one task enhances the performance of another, creating a virtuous cycle of improvement. Accurate intent recognition provides the context needed for precise slot filling, while detailed slot information clarifies the user's intent, resulting in a dialogue system that performs with remarkable accuracy and fluency [18, 166, 212, 232].

As mentioned above, the advantages of joint deep learning models in multi-turn NLU are manifold. Their integrated learning approach, consistency and coherence, efficiency, and improved performance make them the ideal choice for handling complex, multi-turn interactions. As we continue to refine these models, we move closer to creating dialogue systems that can engage users in natural, seamless, and contextually rich conversations, much like an orchestra performing a symphony with impeccable harmony and grace.

Chapter 5
Evaluating Natural Language Understanding

5.1 Overview

We have discussed how understanding of human utterances is represented at different levels of different granularity: that is, intent at the sentence level and slots at the token level. For conversational Natural Language Understanding (NLU), we can add a third tier of domain at the dialogue level.

How should we run experiments that further the ability of machines to perform the task of mapping human utterances to the semantic frame? How should we evaluate how well a system learns to make this mapping?

In this chapter, we will discuss the techniques and issues that arise out of these questions. We will cover the annotation of NLU datasets, quantitative and qualitative evaluation, and experiments to measure the performance. We will discuss the performance of the NLU field on some of the major datasets and the future directions this performance suggests.

5.2 Evaluation Setup

The NLU experiment follows a format very typical in machine learning research. The researcher seeks to address an issue not previously addressed in the literature, or an aspect not explored sufficiently deeply by previous experiments for the field to understand its role. Typically, a new approach, or a new technology, or a new architecture will be brought to bear. A new dataset may have become available, or an existing dataset may have been adapted or extended, which it is believed may help contribute to understanding of the NLU field.

An experiment is then run with the new approach on the dataset. Typically, the headline reported result is the quantitative metric chosen. Implicit here is the null hypothesis that the new approach doesn't perform significantly better on the dataset

S. C. Han et al., *Natural Language Understanding in Conversational AI with Deep Learning*, https://doi.org/10.1007/978-3-031-74364-1_5

than the previous state-of-the-art model, that is the model which has previously recorded the best quantitative result on that dataset. The alternative hypothesis is that the new approach does outperform the benchmark, at a statistically significant level.

It behoves the researcher to design an experiment where the differences to previous approaches are clear and can be modularized. For example, if the experiment proposes both a feature creation step and an NLU architecture that are different to previous approaches, then the research should apportion, via ablation, which step contributes what portion of the improvement in metric observed. In this example, ablation means the experiment is run with feature creation close to or the same as the previous benchmark, and then again with the new feature creation method.

The experiment should be reproducible by others. The best process for this is for the experimental code and datasets to be made available in a public repository. The hardware used and the run times should be reported in research papers.

The above notes are familiar to researchers in machine learning in general, and they will be familiar with the argument that while an arms race in quantitative results is useful, the qualitative aims, achievements, and discussions in the field are just as important. What are the quantitative results important? We aim to produce learning machines that satisfy the needs of their human users at a useful level. If their error rate is too high, human users will shy away from using them. If too often they produce meaningful output, but not to the request the user has provided, the same dissatisfaction will occur.

Qualitative analysis should also be part of an experiment. Why has this new method produced the results that we have observed? Why does the result differ for different datasets? What can we infer from the ablation tests? Has the result illuminated the issue in the field the experiment aimed to explore? What are the weaknesses of the experiment that can be addressed in future research? What future directions for research in the field are now opened up?

Finally, in this era, the researcher must explore ethical responsibility around the resource usage of their experiment and potential biases in the data they use. The use of model cards, or machine learning report cards, now required by major conferences, is a good practice to follow.

5.2.1 The Standard NLU Experiment

In NLU, training is typically performed on annotated utterances; it is a supervised learning experiment. The model creates features from the input utterances and learns to predict an intent and slot labels for each utterance. The loss is a cross-entropy loss calculated for intent and for each slot. A joint loss is typically the sum of these cross-entropy loss, sometimes weighted more or less toward intent or slot. The data is split into a training set and a held-out test set for evaluating performance. The training set is often split further into training and validation sets, especially for deep learning models.

The detail of this experimental setup varies across the literature. Unfortunately, many do not clearly state their setup with respect to dataset organization and hyper-parameters. In those papers which do specify the setup for datasets, most utilized a train-test split, usually as 80%–20%. The training data may be split further to make a validation set, used to monitor generalization performance during training. Results are sometimes reported to be averaged over a number of runs. Alternatively, papers use fivefold or tenfold cross-validation for evaluation.

For parameter tuning, dropout rates ranged from 0.003 to 0.5 and the size of hidden states was normally between 100 and 200, with as low as 64. Some models use the Adam optimizer with a learning rate between 0.0001 and 0.01, or started with 0.02 and halved it after 10 epochs [219], or halved the learning rate when the loss reduced below a threshold [178]. Some papers mention that they set parameters randomly in the beginning, and then apply fivefold validation when tuning parameters. Word embedding dimensions vary from 64 to 1024.

The lack of reporting also occurred with the number of epochs. When stated, most models were trained for less than 50 epochs, with some of these using early stopping. Goo et al. [67] applied 10 epochs for ATIS and 20 epochs for SNIPS in all their experiments and further replicated previous benchmarks using the same etiquette. On the other hand, [72] allowed an unlimited number of epochs with stopping criteria, while [164] used 300 epochs with no early stopping. Further, some models report the final epoch results while others report the best results.

The lack of detail given in many papers leads to difficulties in model result replication, even when code is supplied. This is exacerbated by the saturation of results discussed later in this chapter. Researchers should include sufficient information about setup in their methodology section or at a related public code repository. Again, this is good experimental practice, which is now required by the leading conferences and journals.

5.2.2 Labelling

In the evolution of the domain, the NLU framework relies on supervised learning, specifically employing fully annotated datasets. In this approach, every statement within a dialogue is tagged with at least one intent at the sentence level, and each individual word is marked with a corresponding slot label.

The intent of an utterance typically describes the outcome the speaker desires from the dialogue system. For example, in the ATIS dataset, the sentences “I want to fly from Boston at 838 am and arrive in Denver at 1110 in the morning” and “show me the flights from Baltimore to Oakland” are labelled with intent *atis_flight* as the speaker is seeking flight information. An utterance with a different intent is “show me the fares from Dallas to San Francisco,” which is given intent *atis_airfare*. Some intents from the SNIPS personal assistant dataset are *GetWeather* for “Is it windy in Boston, MA right now?” and *RateBook* for “Give 6 stars to Of Mice and Men.”

For those designing datasets, a consideration for naming intent classes is that some models use word embeddings of the label names in their architecture [283]. Hence, the SNIPS method of using a relevant verb and noun in the intent label may be preferable to the more blunt ATIS style.

Slots are the words in the utterance that carry the crucial semantic details needed to achieve the intent. A single instance of such detail can be carried over multiple words. For example, in the utterance "find me a flight to Las Vegas tomorrow," the destination city consists of the two tokens 'las' and 'vegas' .

As a result of this consideration, NLU uses slot labels in the BIO format [176], where B stands for Beginning, I for Inside, and O for Outside or Other. To be precise, the tagging method used in the NLU task is BIO2 [183], rather than the original BIO. In BIO, a 'B' tag is used only for a new chunk and is immediately preceded by an 'I' tag of the same class. To align with the literature, we will continue to refer to the tagging scheme as BIO.

Token labels are of the form 'B-CLS', 'I-CLS', or 'O', where CLS is an element of a finite set of classes. Then, a span, or chunk, is a contiguous sequence of tokens where each token within is labelled from the same class. The first token in the sequence has the label 'B-CLS', and the finite subsequent tokens only have the label 'I-CLS'. Tokens labelled 'O' do not form spans. Hence, well-formed labellings for a single span are either of the form 'B-CLS' for a single word span or 'B-CLS', 'I-CLS' . . ., and 'I-CLS' for a multi-word span. The label sequence 'B-CLS' and 'B-CLS' consists of two spans of class CLS, while the label sequence 'B-CLS1' and 'B-CLS2' consists of two spans of class CLS1 and CLS2, respectively.

Continuing our example, in the ATIS dataset which covers air travel, the set of slot classes includes *from_loc.city_name* for the city a flight originates from and *to_loc.city_name* for destination cities. Then 'Las Vegas' can be labelled *B-to_loc.city_name, I-to_loc.city_name* if it is a destination in the utterance.

In gold annotation, only well-formed spans can appear, whereas in prediction, malformed spans might appear. For example, an 'I' label immediately following an 'O' label or an 'I' label of one class immediately following a 'B' or 'I' label of another class. Our NLU system should penalize such labellings so they do not occur.

While other schemes for slot labelling exist, for example, introducing M-CLS and E-CLS tags for mid-span and end-of-span tokens, these have not been observed in the NLU literature.

The idea is that slots are the tokens holding the semantic information necessary to successfully fulfill the speaker's intent. However, this doesn't mean they provide all the clues to what that intent is. The values in the slots labelled 'O', that is, the words themselves, are valuable. For example, consider the utterance "find me a flight from Austin to Baltimore tomorrow," which would be annotated ['O', 'O', 'O', 'O', 'O', 'B-from_city', 'O', 'B-to_city', 'B-day'] the tokens 'flight', 'from' and the second occurrence of 'to' give salient clues that the intent is 'atis_flight', especially given the non-'O' slot labels that occur. Perhaps counterintuitive, the values in the 'O' slots here may convey more information useful to intent classification than the

values in the non-'O' slots. That is to say the existence of slot labels 'B-from_city' and 'B-to_city' is useful, their values could be any pair of city names. This is the idea behind delexicalization methods described in the methodologies chapter.

In multi-intent labelling, an utterance may have more than one label. For example, in the ATIS dataset, an utterance like "list for me the times and fares for the morning flights" requests both an airfare (intent atis_airfare) and a flight time (intent atis_flight_time). In the standard annotation, this is given a single merged intent class *atis_airfare#atis_flight_time*. MultiATIS and MultiSNIPS provide examples with up to three separate intents.

Labelling slots with multiple semantic labels has not been observed in the literature, though other types of tags (POS or NER, for example) are sometimes used together with the semantic tag to enable multi-task learning.

Now consider multi-turn conversations where sequences of utterances may be expected to belong to a single topic or domain. The conversation itself may range over a sequence of domains. For example, a user may discuss visiting a tourist attraction over multiple utterances with an agent. Once that is organized, they may have a dialogue where they book a nearby restaurant, and then another section will discuss transport to and from the area. Consider this example from the M2M dataset:

A. 'hi , i want to make a restaurant reservation .'
B. 'okay , where do you want to go , and how many people will there be ?'
A. 'the sushi boat for 6 .'
B. 'for what date ?'
A. 'this friday .'
B. 'at what time ?'
A. '7 pm .'
B. 'so you want a reservation for the sushi boat for this friday at 7 pm for 6 people ?'
A. 'that ' s right .'
B. 'alright . your reservation is confirmed for 6 people at the sushi boat friday at 7 pm'
A. 'thanks !'

The first observation here is that there are two speakers. It is useful to label each utterance with the speaker's name or number, as this aids understanding. Secondly, the intents are qualitatively different for each speaker and, in fact, may consist of two disjoint sets. Depending on the processing you do with this dataset, the intents may look like this:

['GREETING', 'REQUEST', 'INFORM', 'REQUEST', 'INFORM', 'REQUEST', 'INFORM', 'CONFIRM', 'AFFIRM', 'NOTIFY SUCCESS', 'THANK YOU']

Annotation of the domain can occur at the whole dialogue level but more often occurs at the sentence level and consists of a domain drawn from a finite set of labels. This means a model can consider that the domain is likely RESTAURANT and the intent is likely REQUEST so that it may look for probable slot labels like **restaurant name**, **num people**, **date**, etc. In fact, as we previously discussed for the joint intent classification and slot-filling task, each tier should influence the others as the model learns and predicts.

In the M2M dataset, each complete dialogue lies within one domain. Other multi-turn datasets, like MultiWoz, allow change of domain in a single dialogue. A successful dialogue system must learn to identify the shift of domain during a conversation.

5.3 Evaluation Metrics

Now we consider the metrics used to measure performance, starting with intent detection, then slot filling and domain classification.

5.3.1 Intent Classification

Intent Accuracy Detection of intent is a classification task, and the widely used metric of accuracy is used for evaluation. In the NLU context, we thus report the ratio of the number of correct predictions of intent to the total number of utterances. Relatively fewer papers report the error rate, the ratio of wrongly classified samples to the total number of samples, equivalent to accuracy subtracted from 100%, for example, [147].

When utterances have more than one intent label, several approaches are applied to judge a correct prediction. Since they are not doing multiple label detection, most researchers consider the combined label a new label type. Other approaches, as noted in [267], are:

- Count an utterance as a correct intent classification if *any* of the ground truth labels is predicted [117, 126]. This can be used for **single or multi-intent prediction**;

- Count an utterance as a correct intent classification if *all* the ground truth labels are predicted [52, 67]. This is used for **multi-intent prediction**, although it is often not explicitly stated in the literature.

Intent Precision, Recall, and F1 Accuracy can be misleading for datasets with imbalanced classes and is typically not reported on a per-class basis. These limitations can be addressed by using micro-averaged precision, recall, and f1 to evaluate intent prediction [118, 250], though this is rare in the literature. For an intent class C:

- TP_C is the number of True Positives, intents which are correctly classified as of class C.
- FP_C is the number of False Positives, intents which belong to other classes but are incorrectly classified as class C.
- FN_C is the number of False Negatives, intents of class C which are incorrectly classified as other classes.

Two approaches are used: micro-averaged and macro-averaged. In the micro-averaged approach, the TP, FP, and FN are summed across all classes $C \in \mathbb{I}$:

$$Precision = \frac{\sum TP_C}{\sum TP_C + \sum FP_C} \tag{5.1}$$

$$Recall = \frac{\sum TP_C}{\sum TP_C + \sum FN_C} \tag{5.2}$$

In the macro-averaged approach, the precision and recall are computed for each class first, then the average across all n_I classes is reported.

$$Precision = \frac{1}{n_I} \sum_{C \in \mathbb{I}} \frac{TP_C}{TP_C + FP_C} \tag{5.3}$$

$$Recall = \frac{1}{n_I} \sum_{C \in \mathbb{I}} \frac{TP_C}{TP_C + FN_C} \tag{5.4}$$

For both approaches, f1 is computed as:

$$\text{f1} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5.5}$$

A variation used for multi-label identification is precision and recall at the top-k predictions. Here, precision is the ratio of correct labels in the top k predictions divided by k, and recall is the ratio of correct labels in top k predictions over the total number of correct labels [264].

5.3.2 Slot Filling

In slot filling, we are interested in capturing the type of the whole spans containing details relevant to successful intent fulfillment. Slot accuracy, sometimes referred to as word labelling accuracy (WLA) [257], is predictably the ratio of the number of correctly labelled slots to the total number of slots. However, this has little currency as it fails to capture the importance of multi-word spans. The span-based metrics described here both capture this nuance and punish malformed label sequences as described above.

In slot evaluation, we do not consider ground truth 'O' labelled slots. We want to measure how well we predict the annotated semantic frame. That is, for the slot labelling, the non-'O' labels present. Again, this is not to say the values in 'O' slots are not important to understanding.

Span-Based Slot Precision, Recall, and F1 A span is a sequence of words labelled from the same class. For example, the labelling [B-MISC, I-MISC, I-MISC] is a span of class MISC. For a class C, we can thus define at the span level:

- TP_C is the number of spans of class C which are wholly correctly predicted.
- FP_C is the number of spans of a different class which are incorrectly predicted as being of class C.
- FN_C is the number of spans of class C which are incorrectly predicted, partially or wholly, to another class.

Micro-averaged and macro-averaged precision and recall and f1 can then be calculated. In most papers, the slot f1 is reported as the span-based micro-averaged f1 over all classes excluding 'O'. The conlleval.py[1] script is regularly used to calculate this micro-averaged precision, recall, and f1 [43, 47, 75, 128, 231] and is used for the results reported in this book.

Token-Based Slot Precision, Recall, and F1 Token-based slot measures are used in [118], with TP, FP, and FN calculated at the token level, with the same definitions as in Sect. 5.3.1. That is, 'B-CLS' and 'I-CLS' are treated as separate unrelated classes, so some label dependency information captured by the span-based metric is lost. As discussed with slot accuracy, this metric is less useful than the span-based version.

[1] https://github.com/sighsmile/conlleval.

5.3.3 Domain Classification

In the current experimental setup, the domain is classified at the sentence level [71, 115, 190, 231]. As a result, the evaluation metrics are the same as for intent, calculated over the classes in the set $\mathbb{D}$ rather than $\mathbb{I}$.

5.3.4 Semantic Accuracy

In the joint task, we must evaluate the construction of the entire semantic frame. We do this with semantic accuracy which measures what proportion of sentences are correctly mapped to their gold truth semantic frame annotation. A correct mapping is when both the intent and all the slots (including 'O' labels) are correctly predicted. Semantic accuracy is then the number of correctly analyzed sentences divided by the number of sentences. This is easily extended to include the domain in the semantic accuracy when the domain is annotated at the sentence level.

Tests of Significance Several standard tests are used to investigate whether the difference in intent metric between two models is significant. Welch's t-test has been used with the p-value threshold set to 0.05 [58, 60]. Others use the Student's t-test for similar purposes [194]. McNemar's test is also applied to test paired binary classified data to evaluate how well two tests agree with each other. Jeong and Lee [91] used this for the classification of ATIS intents into two domains.

More common is to report the results with a confidence band. Here the experiment is run several times with different initializations, or as folded cross-validation. A band can then be quoted around the mean result with a clearly stated confidence level, typically 95% or 99%. This is both good experimental practice and required by the major conferences and journals.

5.3.5 Other Forms of Analysis

Outside of research that simply attempts to apply a new technology to benchmark datasets, for example, JointBERT [33], qualitative research questions make up part of the contribution. To explore such questions, we need to rely on more than implicit endorsement in the form of increased headline metric performance.

For example, in [243] the authors investigate the use of explicit bidirectional exchange between slot and intent paths for multi-intent datasets. While this had been used for single intent datasets, this approach was new for multi-intent, specifically the use of slot2intent. As well as bidirectional information flow the model uses two global graphs to represent relations between slot semantics and intent labels, and intent semantics and slot labels. The results exceed state of the art, especially in semantic accuracy. An account of the improvement with explanation of the semantic

accuracy boost is given. A full ablation section breaks down the contribution of the different modules and connections in the network. The discussion then gives two examples from the experiment showing the model correcting erroneous predicted labels from after the bidirectional exchange to after the interaction with the knowledge graphs. See Fig. 5.1. This is called example-based explanation.

With these discussions, the reader gains deeper insight into the workings of the model and how it handles the aspects of the joint task that the authors are addressing.

Other qualitative analyses may include observation of patterns in the quantitative results broken down by class. This can lead to interesting hypotheses about the performance of a model based on the class distributions and characteristics.

Similarly, analyzing instances of particular error types can be illuminating—what is common for deletion type errors where non-'O' slot labels are predicted as type 'O'? Or for insertion errors where slots of type 'O' are given another slot label? Are there particular words or syntactic patterns that occur frequently in errors of a certain type? For intents, are there intents that perform particularly poorly? Is this due to their low support in the training or test set, or is there another driver?

Another approach is to use methods from explainable AI (XAI). Explainable NLU seeks to gain insight on what tokens in the input utterance are contributing to the model's output, and what layers in the model architecture are similarly contributing.

Transformer models are suitable for this kind of analysis since they produce multiple attention-based representations aligned with the input tokens. Attention Visualization (also called Attention Maps) looks at the raw attention scores in a model and visualizes them via a heat map. An example is in Fig. 5.2, from [284], shows the attention scores for tokens in an utterance of intent *GetWeather*. In the two models being compared, the lower row has higher attention scores for the words "forecast" and "cold", that a human may also focus on when understanding the request. Analysis of the underlying values can be used for correlation analysis, or ambiguity identification, across the samples in a dataset. Alternatively, the BERTViz tool can be used to show how attention shifts across layers and heads in a Transformer model, a method known as Attention Flow. Creative use of such data is a little explored area of NLU.

Saliency methods aim to find the features, or tokens in NLP, most important to a task's outcome. Layerwise Relevance Propagation (LRP) [10] backpropagates relevance scores through a network where they can be interpreted for the token representations at the initial layer. This method was used for intent classification in [95]. Saliency Maps calculate the gradient of the model output with respect to the input features. A form of numerical derivative analysis is often used where input vectors are perturbed and the effect on the final output is measured. The method of Integrated Gradients [204] is available in Pytorch via the Captum library and has been used for NLP. For example, [51] used it for entailment and question answering. Figure XX from that paper shows a visualization of some sample output where the attribution of each token (sign and strength) to the correct classification of a sentence

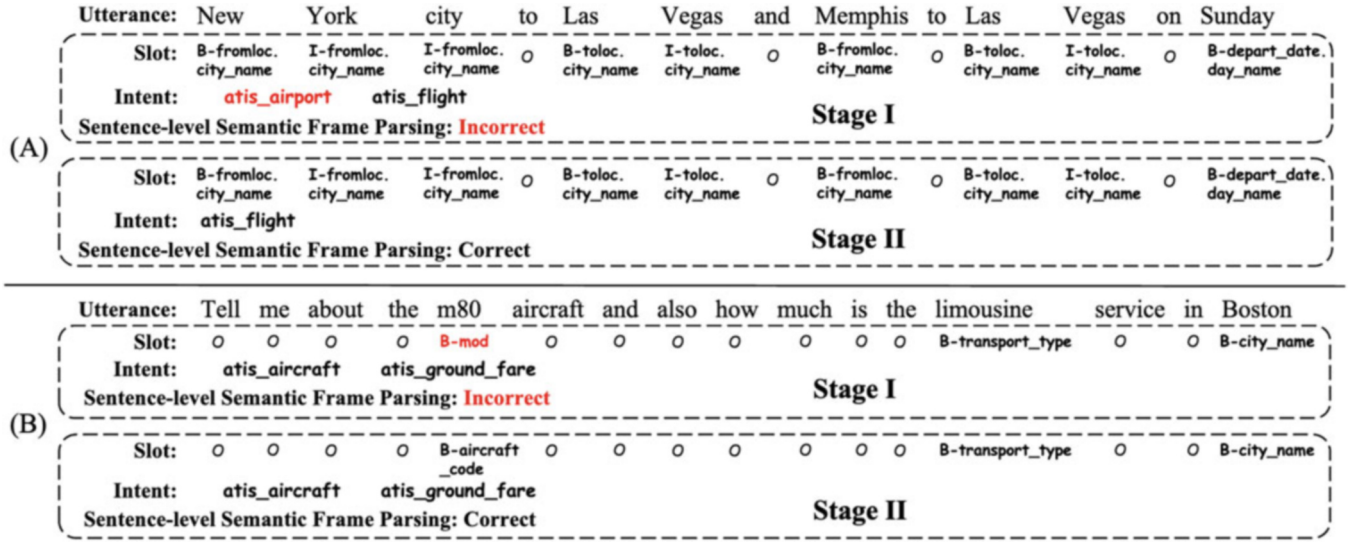

Fig. 5.1 Case study of slot-to-intent guidance (**a**) and intent-to-slot guidance (**b**). Red color denotes error [243]

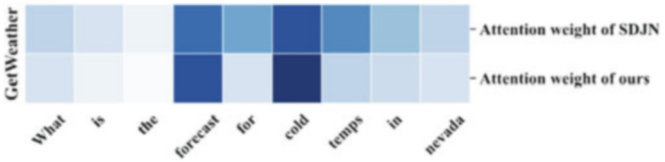

Fig. 5.2 Attention heatmap in different approaches [284]

is visualized. An analysis of benchmark models on the benchmark datasets may extract further understanding of the joint NLU task.

5.4 Performance Trend

The datasets for NLU are discussed in Sect. 3.5. Due to being public and providing a variety of challenges, the ATIS and SNIPS datasets have been used for benchmarking since the advent of deep learning in NLU. To summarize performance in the joint task, we list the models and their reported test results for the ATIS and SNIPS for the standard evaluation metrics (if available) in Table 5.1. Papers are included in this table if at least one of their results is better than the benchmarks reported in papers from the previous calendar year. Interestingly, this qualification means no surveyed papers from after 2021 are included in the table.

Several interesting patterns can be observed based on the results available: (1) The overall improvement in slot F1 (from 87.3 in 2016 to 98.78 in 2019) and semantic accuracy (from 73.2 in 2016 to 93.6 in 2020) for SNIPS over time is much more significant than ATIS (from 93.96 in 2014 to 98.75 in 2019 for F1 and from 78.9 in 2016 to 93.1 in 2021 for semantic accuracy), while intent accuracy moves in the opposite way (from 98.32 in 2016 to 99.76 in 2019 for ATIS and from 96.7 in 2016 to 99.98 in 2019 for SNIPS). (2) For those models that reported slot F1 and intent accuracy on both datasets, 14 out of 28 performed better in slot F1 for ATIS and in intent accuracy for SNIPS. (3) Before 2019, the best performance for semantic accuracy came from ATIS, while a shift to SNIPS starts from 2019 ending with the best semantic accuracy in that dataset.

These patterns point to ATIS being easier to classify, probably due to its majority intent class, single domain, smaller vocabulary and lack of variety in utterance structure. As newer or more sophisticated architectures are designed for better generalizability, the performance on SNIPS, the more complex dataset, improves.

The number of epochs used to derive reported performance in NLU models for comparison with others varies widely. In 2018, [67] made a performance comparison under the same epoch settings. They used 10 epochs for ATIS and 20 epochs for SNIPS and replicated the models of earlier benchmarks using this

Table 5.1 NLU performance on ATIS and SNIPS-NLU datasets (%). * denotes ATIS 10 epoch and SNIPS 20 epoch, i denotes epoch count implied, † indicates GitHub available, – denotes not reported. Hakkani-Tür et al. [74] and Liu and Lane [126] were reported in [67]

	ATIS			SNIPS		
Paper, Model	Slot(F1)	Int.(Acc)	Sem.(Acc)	Slot(F1)	Int.(Acc)	Sem.(Acc)
[91] Joint 2	94.42	93.07	–	–	–	–
[245] CNN TriCRF	95.42	94.09	–	–	–	–
[71] RecNN+Viterbi	93.96	95.4	–	–	–	–
[190] RNN Joint + NE	96.83	95.4	–	–	–	–
[74]*	94.3	92.6	80.7	87.3	96.9	73.2
[34] K-SAN Syntax	95.38	–	84.32	–	–	–
[272] W+N	96.89	98.32	–	–	–	–
[126]*	94.2	91.1	78.9	87.8	96.7	74.1
[67] Full Att.* †	94.8	93.6	82.2	88.8	97.0	75.5
[67] Intent Att.* †	95.2	94.1	82.6	88.3	96.8	74.6
[226] Atten.+aligned	97.76	97.17	–	–	–	–
[56]	98.02	98.43	–	–	–	–
[117] †	96.52	98.77	–	–	–	–
[116]	94.81	98.54	–	–	–	–
[225] †	96.89	98.99	–	–	–	–
[260] ACJIS Model	96.43	98.57	–	–	–	–
[194] ELMo	95.62	97.42	87.35	93.9	99.29	85.43
[52] SF 1st+CRF i †	95.8	97.8	86.8	91.4	97.4	80.6
[265] Capsule i †	95.2	95.0	83.4	91.8	97.3	80.9
[72] CNN3L+5kern	96.95	98.36	–	94.22	99.1	–
[32]	96.54	98.91	–	93.94	99.71	–
[268]	95.1	97.2	–	93.3	98.9	–
[43] BiLSTM-CRF	95.6	96.6	86.2	94.6	97.4	87.2
[128] CM-Net with GloVe	96.2	99.1	–	97.15	99.29	–
[128] CM-Net with BERT	–	–	–	97.31	99.32	–
[164] †	95.9	96.9	86.5	94.2	98	86.9
[164] Model+BERT	96.1	97.5	88.6	97	99	92.9
[60] HCNN/CRF	97.32	99.09	–	94.38	98.24	–
[26]	95.7	97.8	88.2	96.2	99	91.6
[275]	98.75	99.76	–	98.78	99.98	–
[161] Base	95.4	96.1	–	94.8	98	–
[161] Base+BERT	95.8	96.6	–	94.5	97.6	–
[33] BERT †	96.1	97.5	88.2	97	98.6	92.8
[33] BERT+CRF †	96	97.9	88.6	96.7	98.4	92.6
[221] SASGBC	96.69	98.21	91.6	96.43	98.86	92.57
[205] fully-E@EMG-CRF	96.4	99.0	89.6	97.2	99.7	93.6
[75]	96.3	98.6	88.6	97.2	99.2	92.8
[87]	96.4	98.2	88.5	97.6	99.3	93.0
[165] BERT	97.1	98.8	93.1	96.1	98.0	88.8
[207]	97.3	98.3	90.2	98.3	98.9	90.2

Table 5.2 NLU performance on ATIS and SNIPS-NLU datasets (%) using ATIS 10 epoch and SNIPS 20 epoch (i denotes epoch count implied). † denotes reproduced by this book, – denotes not reported

	ATIS			SNIPS		
Paper, Model	Slot(F1)	Int.(Acc)	Sem.(Acc)	Slot(F1)	Int.(Acc)	Sem.(Acc)
[74]	94.3	92.6	80.7	87.3	96.9	73.2
[126]	94.2	91.1	78.9	87.8	96.7	74.1
[67] Full Att	94.8	93.6	82.2	88.8	97.0	75.5
[67] Int. Att	95.2	94.1	82.6	88.3	96.8	74.6
[117] †	94.82	97.00	84.00	88.93	97.71	76.43
[225] †	95.14	96.08	84.87	88.46	96.71	75.39
[52] SF 1st+CRF i	95.8	97.8	86.8	91.4	97.4	80.6
[265] Capsule i	95.2	95.0	83.4	91.8	97.3	80.9
[72] CNN3L,5 kern	95.27	97.37	–	92.3	97.57	–
[164] †	93.15	95.9	80.4	90.88	97.14	79.71
[33] BERT †	95.54	97.54	87.35	96.91	98.43	92.43
[33] BERT/CRF †	96.03	97.76	88.47	96.60	98.57	92.14
[75] Bi-ED	**96.3**	**98.6**	**88.6**	**97.2**	**99.2**	**92.8**

number of epochs. Alternatively, [72] clearly specified the epochs to converge for each of the models being compared. We consider these as two alternative methods for comparing performance in terms of epochs and propose that they can offer an additional perspective in model comparison and selection for future research. In Table 5.2, we follow [67] and provide the test results for those models that use 10 epochs for ATIS and 20 epochs for SNIPS. These results are either from papers who also follow this etiquette, or from the reproduction by [67], or are produced by using the GitHub code supplied by the authors and using the required number of epochs (indicated by ††). In the latter case, we can also confirm the consistent calculation of intent and semantic accuracy as well as span-based slot F1, as this is often not specified by the original authors. As seen in the table, similar patterns can be observed when we specify the number of epochs: that a significant improvement in slot F1 and semantic accuracy for SNIPS has been made since 2019 and that most of the models perform better in slot F1 for ATIS and intent accuracy for SNIPS. These patterns may be related to the various distributions of slot and intent labels and the different nature of domains of the two datasets as addressed by different architectures.

The results are now high enough for the two most used datasets that any fruitful new developments may be lost in results that appear to not significantly increase the metrics for these datasets. That no surveyed papers after 2021 have improved on results prior emphasizes this. It is desirable that new datasets should become part of the NLU reporting canon. A surge in papers in 2022 and beyond addressing the multi-intent problem covered by the mixATIS and mixSNIPS datasets point to one direction to progress. Similarly, the language generalizability and transfer learning

Table 5.3 Results on three SLU benchmark datasets [282]

	ATIS		SNIPS	
Model	Intent (Acc.)	Slot (F1)	Intent (Acc.)	Slot (F1)
Fine-tuned SOTA	98.54	96.46	99.12	97.21
GPT 3.5 (text-davinci-003)	90.03	55.72	98.00	68.90
ChatGPT	75.22	15.71	97.71	58.24
GPT-SLU	88.90	67.04	98.50	75.65

issues are being addressed with MultiATIS++. These approaches mean some new problems are addressed—multiple intent and transfer learning across languages. While useful we must ask if these directions extend to a more rounded overall understanding of natural language.

One limitation of the existing benchmarks is their single-turn nature. A direction for attention must be multi-turn datasets specifically annotated for NLU as this is a more natural environment for conversational NLU. Then another issue of the existing benchmarks is the closed domain and finite intent and slot sets. New datasets should address the issues of unlabelled data and identification of emerging domains and slots.

These last two issues are of interest to recent developments in chatbot interfaces to LLMs. Consider our example in Chapter X where we, the human NLU user, propose a problem to an LLM, then ask for improvements or make suggestions—a multi-turn conversation. The reader can consider their own interactions where the domain of this conversation changes as they use the chatbot during a day's work. So how do LLMs perform in the multi-tier NLU tasks?

Recent papers have addressed this question. In [282], zero-shot joint task NLU was performed on ATIS and SNIPS. The SOTA they benchmarked to from [30] were surveyed by us but not included in the performance table above as they did not improve on previous results. Their GPT-SLU results improved on previous work using GPT-3.5 and ChatGPT (denoted by * and reported in [158]). It is clear that the zero-shot performance does not match the performance of models specifically trained for NLU. This is especially clear for slot tagging (Table 5.3).

However, there are better approaches for using LLMs for tasks they haven't been trained on. In context learning, for example, gives examples of the task in the prompt along with an unanalyzed sample for the model to work on. The ILLUMINER model of [145] uses this approach, and we reproduce their result tables below (Tables 5.4 and 5.5).

Considering the results for SNIPS, we note again that the result for intent accuracy and slot filling F1 still fall short of the benchmark for few shot in context learning. The few-shot improvement over zero shot is noticeable for all datasets used in the study. The best results are for a LORA adaptation to the existing LLM. In a LORA approach, the existing LLM weights are frozen, and a smaller model is trained in parallel to the LLM on the specific new task of interest. Hence, this becomes another example of supervised training on the NLU task, and it is perhaps

Table 5.4 Intent accuracy in zero-shot, few-shot, and LoRA settings, across different datasets and InstructLLMs [145]

		SNIPS			MASSIVE			MultiWoz		
Instruct-LLM	Size	zero-	few-	LoRA	zero-	few-	LoRA	zero-	few-	LoRA
falcon-7b-Instruct	7B	0.301	0.570	0.779	0.103	0.360	0.546	0.558	0.748	0.941
bloomz-7b1	7B	0.795	0.686	0.930	0.265	0.435	0.657	0.899	0.894	0.941
flan-t5-xxl	11B	**0.937**	**0.940**	**0.962**	**0.726**	**0.741**	**0.825**	**0.973**	**0.982**	**0.972**
vicuna-13b-v1.5	13B	0.574	0.920	0.950	0.333	0.688	0.759	0.425	0.977	0.972
WizardLM-13B-V1.1	13B	0.720	0.674	0.921	0.355	0.678	0.731	0.962	0.933	0.956

Table 5.5 Slot filling F1 in zero-shot, few-shot and LoRA settings, across different datasets and InstructLLMs [145]

		SNIPS			MASSIVE			MultiWoz		
Instruct-LLM	Size	zero-	few-	LoRA	zero-	few-	LoRA	zero-	few-	LoRA
ILLUMINER: Single-prompt IE										
falcon-7b-Instruct	7B	0.136	0.543	0.835	0.042	0.421	0.585	0.319	0.640	0.928
bloomz-7b1	7B	0.177	0.541	0.876	0.043	0.349	0.640	0.278	0.527	0.943
flan-t5-xxl	11B	**0.310**	0.647	**0.909**	**0.125**	0.473	**0.735**	0.462	0.753	0.945
vicuna-13b-v1.5	13B	0.222	0.554	0.908	0.103	0.369	0.724	**0.500**	**0.859**	**0.957**
WizardLM-13B-V1.1	13B	0.298	**0.685**	0.899	0.116	**0.474**	0.710	0.428	0.830	0.951
Multi-prompt IE										
flan-t5-xxl	11B	**0.221**	0.380	**0.904**	**0.058**	0.195	**0.658**	**0.360**	0.547	**0.933**
vicuna-13b-v1.5	13B	0.111	**0.569**	0.755	0.021	**0.252**	0.597	0.146	**0.798**	0.889
WizardLM-13B-V1.1	13B	0.127	0.531	0.703	0.020	0.202	0.509	0.255	0.717	0.855

not surprising this gives better results than the zero shot and few shot in context methods.

Again, we return to our river crossing example and recall there that while the intent (and indeed domain) of the prompt was identified well, it was the slots important to the fulfillment of the intent that wasn't fully identified. LLMs in their current state have issues performing NLU in their own actions, let alone when asked to perform NLU as a prompted task. The interfaces here promise an exciting area for future research.

Chapter 6
Applications and Case Studies in Natural Language Understanding

6.1 Overview

As the digital age unfolds, Natural Language Understanding (NLU) is at the cutting edge of technology, becoming a part of our everyday lives. This chapter explores the many areas where NLU is making a difference, showing how it is used in real-world applications.

Imagine a world where machines understand and respond to human language with the same nuance as a skilled conversationalist. This is not a distant future but our present, thanks to the rapid progress in NLU technologies. From financial analysts interpreting market trends to doctors diagnosing patients through clinical notes, NLU turns data into useful insights across many fields.

NLU bridges the gap between what humans mean and how machines respond, making technology easier to use and more responsive. In finance, NLU uncovers market trends and helps guide investments. In healthcare, it reads clinical notes to aid early diagnosis and create personalized treatment plans. In the legal field, it helps with research, making sure justice is served accurately. NLU makes industries better by turning complex data into easy-to-understand information.

The journey to perfect understanding is not without challenges. Language can be unclear, context can be subtle, and human expression is diverse. But these challenges drive innovation. Advances in machine learning, smart algorithms, and growing data are pushing NLU forward, bringing us closer to machines that truly understand.

In this section, we will see how important NLU is in different areas. We will look at real examples and case studies to show NLU's practical effects. We will explore its role in finance, healthcare, and the legal sector, showing how NLU changes industries and creates new possibilities.

S. C. Han et al., *Natural Language Understanding in Conversational AI with Deep Learning*, https://doi.org/10.1007/978-3-031-74364-1_6

By looking at these applications, we will understand how NLU is used in various sectors, driving new ideas and better results. This highlights NLU's potential and inspires further progress in the field.

6.2 Financial Applications in NLU

In the bustling heart of the financial district, where skyscrapers pierce the clouds, and the hum of trading floors reverberates, NLU is quietly revolutionizing the industry. The world of finance, with its ceaseless flow of information, is a realm where precision and speed are paramount. Here, NLU emerges as a silent partner, tirelessly working behind the scenes to decode the language of money.

The financial sector, a bastion of human expertise and intuition, now intertwines itself with advanced technology. NLU is at the forefront of this transformation, empowering institutions to navigate the complex tapestry of data that defines modern finance. From the cryptic musings of market analysts to the nuanced sentiments of global news, NLU systems sift through vast amounts of text, extracting valuable insights that drive decision-making processes.

Consider the scenario of a financial analyst, eyes glued to multiple screens, each brimming with charts, news feeds, and reports. Amid this deluge of information, patterns and trends often lie hidden, waiting to be uncovered. Here, NLU steps in, parsing through the noise, identifying significant signals, and presenting them in a coherent and actionable format. It is as if the analyst now has an invisible ally capable of interpreting the intricate language of the financial world with uncanny accuracy.

6.2.1 Financial Application Case Studies

A Vanguard Assistant (AVA) A Vanguard Assistant (AVA), a task-oriented chatbot, is designed to support phone call agents while interacting with clients on live calls. With AVA, the consultation process between phone agents and experts is transformed into a seamless end-to-end conversational AI system.

The integration with chatbot is aimed to revolutionize the client consultation experience. The primary objective is to significantly reduce operating costs by reducing call holding times and diminishing the dependency on human experts. In doing so, they aspire to transform client interactions with NLU methods, fostering an environment where clients can increasingly manage their needs autonomously within a controlled framework. The success of this venture hinges on the NLU component's ability to accurately understand client intents and promptly escalate matters that require human intervention.

Figure 6.1 presents a system overview of AVA, which gradually replaces traditional human-human interactions between phone agents and internal experts and eventually allows clients to self-provision interactions directly with the AI system. Now, phone agents interact with AVA chatbots deployed on Microsoft Teams on the company Internet, and their questions are preprocessed by a sentence completion model to correct misspellings.

Banking77 The sensitive information shared during sessions necessitates stringent security measures. Task-oriented chatbots like AVA often need access to critical internal systems and confidential data to accomplish specific tasks effectively. Consequently, industries prefer 100% on-premise solutions over cloud-based alternatives. These on-premise systems allow for complete customization and rigorous monitoring, ensuring smooth integration while safeguarding sensitive information.

In financial services, the ability to accurately classify customer service intents is paramount. One particularly compelling study leveraged the **Banking77** dataset [24], a real-life collection of customer service intents and their classification labels. This dataset stands out in the intent detection literature for its focused scope within a single domain, encompassing a substantial number of labels—seventy-seven, to be precise, as shown in Table 6.1. The nuanced nature of these classes, with many exhibiting tight overlaps, makes Banking77 an ideal candidate for practical business applications.

Previous research has tackled various challenges associated with this dataset. For instance, efforts have been made to address labelling errors. Other studies have explored innovative methods for pre-training intent representations, such as the work by [120]. While effective, these approaches often demand a high level of technical expertise, posing a barrier to broader adoption.

The quest to enhance intent classification in financial services continues to evolve. By leveraging datasets like Banking77, researchers and practitioners alike are striving to refine the accuracy and efficiency of virtual agents (VAs). These advancements promise to transform customer interactions, making them more seamless and intuitive and ultimately reshaping the landscape of financial service delivery.

6.2.2 Financial Open Intent Classifier

With the recent surge in NLP technologies within the financial domain, banks and other financial entities have increasingly adopted VAs to assist their customers. One of the most challenging problems for these VAs is determining the user's intent, especially when the intent is unseen or open during the VA's training. This challenge is particularly evident in scenarios where user utterances, such as those from the Banking77 dataset [24], introduce intents that were not known during the training phase. For instance, while intents like Card Arrival and Exchange Rate

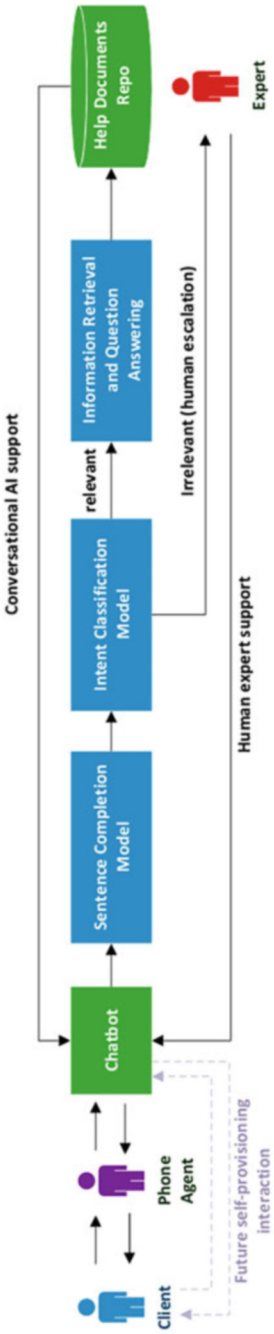

Fig. 6.1 End-to-end conceptual diagram of AVA [259]

Table 6.1 Banking77 dataset statistics for training and testing sets

Banking77 Statistics	Train	Test
Number of examples	10,003	3,080
Minimum length in characters	13	13
Average length in characters	59.5	54.2
Maximum length in characters	433	368
Minimum word count	2	2
Average word count	11.9	10.9
Maximum word count	79	69
Number of intents	77	77

Table 6.2 Sample utterances and their labels in Banking77 dataset

Utterance	Label
When will I get my card?	Card Arrival
What exchange rates do you offer?	Exchange Rate
My card hasn't arrived yet.	Card Arrival
Is it a good time to exchange?	Exchange Rates
Is it possible to get a refund?	**Open**
Why has my withdrawal not posted?	**Open**

might be familiar, requests related to refunds and withdrawals present as open intents (Table 6.2).

For instance, most traditional systems demonstrated a session where a financial VA adeptly responded to a user's query for investment advice. Similarly, CalFE has utilized commercial chatbot frameworks to train finance-specific VAs, as discussed by [99]. The impact of a VA's social presence on user engagement was also evaluated by [150], highlighting the necessity of accurately extracting intents from user utterances to enhance interaction quality.

In the study, [120] illustrated that pre-training intent representations could significantly improve the clarity of decision boundaries, thereby enhancing performance in intent classification tasks within the financial domain. Their approach involved prefixing and fine-tuning the last layer of a LLM. This method yielded impressive results, achieving an accuracy of 82.76% and a Macro-F1 Score of 87.35% on the Banking77 benchmark in the FullData text classification setting.

These advancements underscore the transformative potential of NLU technologies in the financial sector. By improving the ability to classify and respond to user intents, virtual agents are not only enhancing customer service but also paving the way for more efficient and intelligent financial operations.

6.2.3 User Log-Based Intent Classification

In practical financial scenarios, as the number of intents proliferates, the challenge intensifies. It becomes imperative to identify utterances corresponding to known

intents from a vast collection of unlabeled client conversation samples, utilizing only a few labelled examples as reference points. Concurrently, there is a pressing need to unearth new, previously unknown intents from the remaining unlabeled utterances.

As shown in Fig. 6.2, there are two dual objectives. Firstly, they focus on detecting known intent utterances within a large pool of unlabeled samples, leveraging a limited set of labelled instances. Secondly, they discover new, unknown intents from the residual unlabeled data. By addressing these tasks, they aim to enhance the adaptability and accuracy of dialogue systems in handling an ever-evolving array of user intents.

This nuanced approach not only fortifies the dialogue systems' capability to manage a growing and diverse set of user intents but also ensures that the systems remain responsive and relevant in the face of changing user needs and linguistic patterns. The continuous refinement and discovery of intents underpin the efficacy of conversational agents, particularly in the financial sector, where understanding and promptly responding to customer queries is crucial.

6.2.4 NLU in Opinion Building Domain

In the argumentative dialogue systems within information-seeking and opinion-building, a sophisticated NLU framework emerges as a vital tool.

The NLU systems in opinion building [3] comprise two integral sub-models: an intent classifier and an argument similarity model. The intent classifier is tasked with discerning the underlying purpose behind user utterances, categorizing them into predefined intents with remarkable accuracy. Meanwhile, the argument similarity model evaluates the coherence and relevance of arguments presented during dialogues, ensuring that the system can intelligently navigate and contribute to discussions.

Separate evaluations were conducted using the publicly available Banking77 and STS benchmark datasets to rigorously assess the efficacy of these sub-models. These datasets, renowned for their robustness, provided a comprehensive testing ground for the framework, highlighting its potential to revolutionize argumentative dialogue systems.

By integrating such advanced NLU capabilities, the framework not only facilitates more nuanced and insightful interactions but also significantly enhances the system's ability to engage in meaningful opinion-building dialogues. This development holds particular promise for the financial sector, where clear, logical, and persuasive communication is paramount.

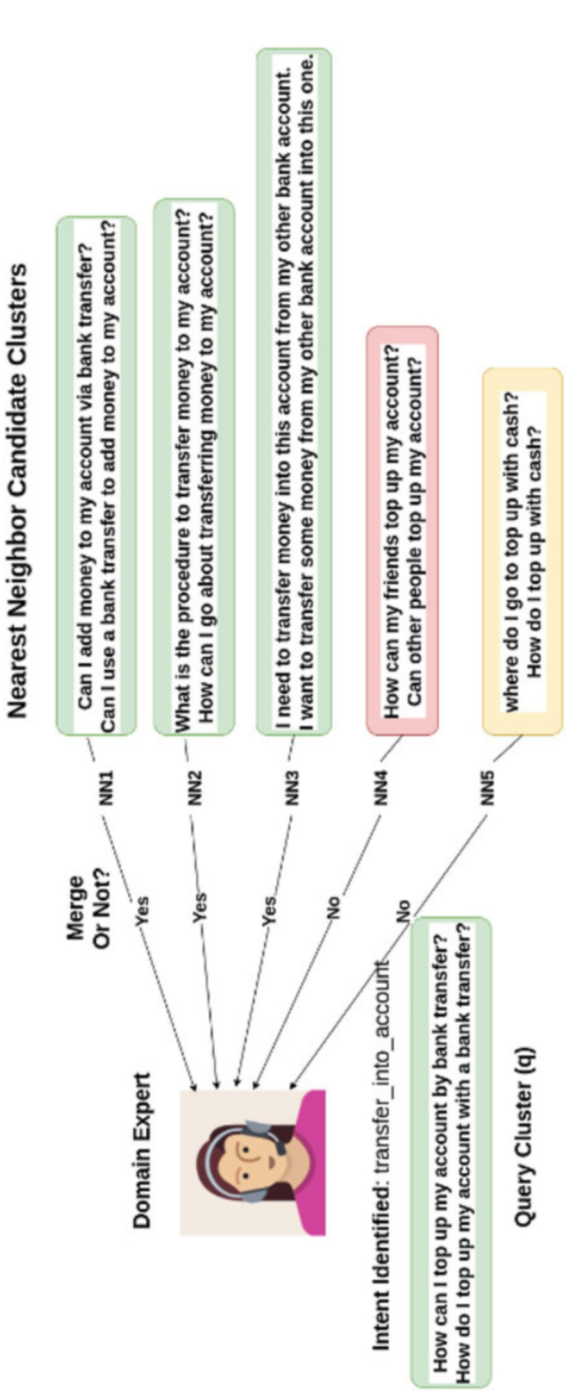

Fig. 6.2 Cluster Merger Algorithm: In each iteration, a Domain Expert is shown (s = 5). Candidate Clusters, which are nearest neighbors of Query Cluster (q) where per cluster, only (p = 2) utterances are shown to the expert. Based on the reply from the domain expert, some candidate clusters are merged into the query cluster, and the cluster definition is updated [111]

6.2.5 Conclusion

As we conclude our exploration of Financial NLU and Conversation Applications, we stand on the precipice of a transformed financial landscape. Integrating Natural Language Understanding (NLU) technologies has ushered in a new era of more intuitive, efficient, and responsive client interactions. From the sophisticated capabilities of AVA, the Vanguard assistant, to the meticulous classification of intents using the Banking77 dataset, we have witnessed how these advancements reshape the sector.

The journey through this section has highlighted the profound impact of NLU in enhancing client experiences and streamlining financial operations. The innovative techniques for pre-training intent representations and discovering new intents underscore the dynamic nature of NLU, ensuring that conversational agents remain adept at navigating the ever-evolving complexities of human language.

Moreover, the introduction of argumentative dialogue systems in the opinion-building domain opens new vistas for engaging and persuasive interactions. By seamlessly integrating intent classification and argument similarity modelling, these systems promise to elevate the quality of dialogues, making them more meaningful and impactful.

As we turn the page to the next section, we carry forward the insights gained from the financial sector into the realm of Medical NLU and Conversation Applications. Here, the stakes are higher, and the potential for NLU to revolutionize healthcare is immense. We will delve into how these technologies are transforming patient interactions, aiding in diagnosis and treatment, and ultimately enhancing the quality of care.

6.3 Medical Applications in NLU

In the sterile, hushed corridors of hospitals and clinics, where lives hang in the balance every second, the integration of NLU in conversation systems heralds a new dawn in healthcare. Medicine, an art as much as a science, thrives on the precise exchange of information. Here, in this world of intricate diagnoses and nuanced patient histories, NLU emerges as a beacon of innovation, transforming how we perceive and deliver medical care.

The medical domain, with its vast repositories of knowledge and ever-growing databases of patient records, presents a formidable challenge for practitioners. Amid the flurry of activity, where doctors and nurses work tirelessly to decode the mysteries of the human body, NLU steps in as a crucial ally. It sifts through oceans of data, making sense of clinical notes, lab reports, and patient interactions, enabling healthcare professionals to make informed, timely decisions.

Imagine a bustling emergency room where doctors are inundated with critical cases, and every moment is precious. Here, NLU-powered conversation systems can analyze patient symptoms, cross-reference medical histories, and suggest possible diagnoses with remarkable speed and accuracy. It is as if the physician now has an omnipresent assistant capable of piecing together complex medical jigsaw puzzles with uncanny precision.

Beyond the confines of emergency care, NLU's applications permeate various facets of medicine. In the realm of patient-doctor interactions, conversational agents equipped with NLU provide round-the-clock support, answering patient queries and offering medical advice. These virtual assistants not only alleviate the burden on healthcare staff but also ensure that patients receive timely and accurate information, fostering a sense of reassurance and trust.

Furthermore, it extends to the analysis of Electronic Health Records (EHRs). By meticulously parsing through these records, those systems can identify patterns and trends that may elude even the most experienced clinicians. This capability is instrumental in early disease detection, where subtle indicators, often buried in voluminous data, can signal the onset of conditions that require prompt intervention.

In the medical field, we will explore case studies that highlight its transformative impact. From patient-doctor interaction systems to clinical decision support, each example will underscore how NLU is reshaping the healthcare landscape.

6.3.1 Medical Application Case Studies

ELIZA: The Pioneer of Conversational AI in Medical Domain In the history of artificial intelligence, **ELIZA** [230] holds a special place as one of the earliest examples of a natural language processing system. Developed in the mid-1960s (1966) by Joseph Weizenbaum at the MIT Artificial Intelligence Laboratory, ELIZA was designed to simulate conversation by recognizing keywords and responding with pre-determined rule-based NLU scripts. Though simplistic by today's standards, ELIZA's most famous script, the "DOCTOR" script, mimicked the conversational style of a Rogerian psychotherapist.

ELIZA's ability to ask probing questions and reflect user inputs back as part of its responses created an illusion of understanding, which, while rudimentary, laid the groundwork for future advancements. In a medical context, ELIZA's approach highlighted the potential for AI to assist in therapeutic settings, albeit in a very limited manner.

PARRY: A Step Toward Simulated Paranoia Building upon the foundations laid by ELIZA, psychiatrist Kenneth Colby developed **PARRY** [38] in the early 1970s, 1972 . PARRY represented a significant leap forward in the complexity of conversational agents. While ELIZA mimicked a non-directive therapist, PARRY was designed to simulate the thinking of a person with paranoid schizophrenia. This

was achieved through a model that included rules for generating responses based on an internal state that could be influenced by interactions with users.

PARRY's ability to simulate a specific mental disorder was a pioneering effort for psychiatric research. It demonstrated how conversational AI with rule-based NLU components could be employed to deepen our understanding of mental health conditions and develop therapeutic strategies. PARRY's interactions were more coherent and contextually aware than ELIZA's, providing a more realistic simulation of human-like conversation and showcasing the potential for AI to contribute to medical research and diagnostics.

IBM Watson: Revolutionizing Medical NLU IBM Watson gained fame in 2011 when it competed on the quiz show "Jeopardy!" and won against human champions. However, its capabilities extend far beyond trivia. In the medical field, IBM Watson has been utilized to analyze vast amounts of medical literature, patient records, and clinical data to assist doctors in diagnosing and treating patients [89]. Watson's ability to understand natural language and generate insights from unstructured data marks a significant advancement in NLU technology. By integrating with electronic health records and continuously learning from new medical data, Watson provides evidence-based recommendations, helping to improve patient outcomes. It can process and analyze data much faster than any human, identifying patterns and correlations that might be missed by even the most diligent practitioners. This makes Watson an invaluable tool in personalized medicine, offering tailored treatment options based on the latest research and individual patient profiles.

Watson's ability to understand natural language and generate insights from unstructured data marks a significant advancement in NLU technology. By integrating with electronic health records and continuously learning from new medical data, Watson provides evidence-based recommendations, helping to improve patient outcomes. It can process and analyze data much faster than any human, identifying patterns and correlations that might be missed by even the most diligent practitioners.

As we develop and refine these technologies, the promise of AI in medicine becomes ever more tangible, heralding a future where AI-driven dialogue systems play a crucial role in medical diagnosis, treatment, and patient care.

6.3.2 Health Information Chatbot and NLU

In healthcare, where timely and accurate information is crucial, **SHIHbot** [20] emerges as a pioneering solution. Deployed on Facebook, SHIHbot is designed to answer a wide variety of sexual health questions related to HIV/AIDS. This innovative chatbot leverages a meticulously compiled response database drawn from professional medical and public health resources, ensuring users receive reliable and trustworthy information.

The backbone of SHIHbot's functionality is NPCEditor, a sophisticated response selection platform. NPCEditor employs a statistical classifier trained on a corpus of linked questions and answers. This classifier, upon receiving a new user utterance, ranks all available responses to select the most appropriate one. To the best of our knowledge, SHIHbot represents the first retrieval-based chatbot of its kind deployed on a large public social network. It highlights the ability of NLU technologies to bridge the gap between medical expertise and public accessibility, offering a glimpse into the future of healthcare communication.

Quro: Revolutionizing Symptom Checking Using NLU **Quro** [64], a personalized chatbot-oriented dialogue system, exemplifies this potential by engaging patients in meaningful conversations about their symptoms. Through a seamless and intuitive dialogue, Quro gathers information about users' symptoms and provides a personalized pre-synopsis based on their individual profiles.

One of the significant advantages of Quro is its ability to perform simple symptom analysis through an NLU-based conversation framework, eliminating the need for cumbersome form-based data entry. This approach not only enhances user experience by making the process more engaging and less tedious but also ensures that critical information is captured accurately and efficiently. By streamlining the initial stages of medical assessment, Quro can potentially reduce the burden on healthcare providers and expedite getting patients the care they need.

MamaBot: Natural Language Understanding Support for Women and Families during Pregnancy In the ever-evolving healthcare landscape, **MamaBot** [215] emerges as a beacon of support for pregnant women, mothers, and families with young children. This system harnesses the power of ML and NLP to provide timely and personalized assistance to its users. Developed using the Microsoft Bot Framework and Language Understanding Intelligent Service (LUIS), MamaBot exemplifies the technologies to meet the specific needs of patients and their families.

MamaBot is a virtual support system that understands and responds to the diverse requirements of pregnant women and mothers. By applying NLU techniques, it interprets user queries and delivers relevant information and instructions. This capability is particularly valuable in providing guidance during various stages of pregnancy and early childhood, offering reassurance and practical advice in real time.

The application scenario for MamaBot is user-centric as shown in Fig. 6.3. It is designed to deliver support in various situations, from routine pregnancy care to urgent queries that may arise unexpectedly. By engaging in meaningful multi-turn conversations, MamaBot helps users navigate the complexities of pregnancy and childcare with confidence. Its ability to process natural language inputs ensures that interactions are intuitive and user-friendly.

CARO: Empathetic Conversation Understanding and Advice for Major Depression In the mental health, CARO [77] stands as a remarkable innovation, offering empathetic conversations and medical advice for individuals suffering from major depression. This chatbot is designed to provide not just information

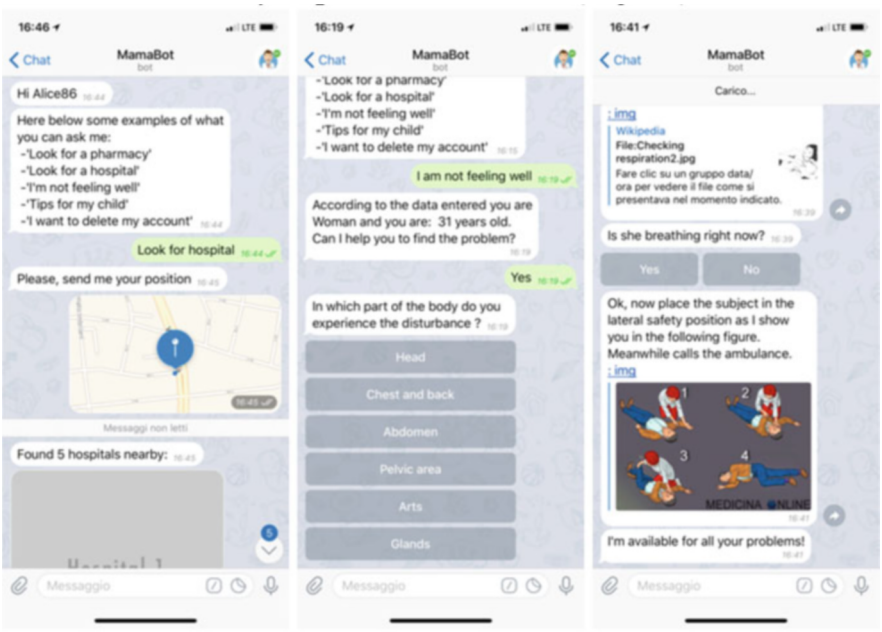

Fig. 6.3 MamaBot use cases [215]

but also emotional support, addressing both the intellectual and emotional needs of its users. CARO's unique capability lies in its ability to sense the conversational context, understand user intent, and interpret associated emotions. Through NLU techniques, CARO can engage in meaningful dialogues, offering comfort and guidance to those in need. This empathetic approach ensures that users feel heard and understood, which is crucial in managing conditions like major depression.

The chatbot's ability to detect and respond to a conversation's emotional undertones is particularly valuable. By recognizing signs of distress or anxiety, CARO can adjust its responses to be more supportive and comforting. This not only helps provide immediate relief but also builds a trusting relationship between the user and the chatbot, making it easier for individuals to open up about their struggles. CARO's integration of NLU extends beyond basic interactions; it can provide personalized medical advice tailored to each user's specific needs. This includes recommending coping strategies, suggesting lifestyle adjustments, and even advising when to seek professional help.

Med-PaLM: Large Language Models for User Language Understanding and Clinical Knowledge In the ever-evolving landscape of healthcare, the integration of large language models has opened new frontiers. **Med-PaLM** [195] stands at the forefront of this revolution, demonstrating the profound capabilities of large language models in encoding and utilizing clinical knowledge. This groundbreaking

approach is encapsulated in the **MultiMedQA** [195] benchmark, a comprehensive evaluation framework that combines six existing medical question-answering datasets with a newly introduced dataset, **HealthSearchQA** [195]. The MultiMedQA benchmark amalgamates diverse datasets, including **MedQA** [92], **MedMCQA** [156], **PubMedQA** [93], **LiveQA** [1], **MedicationQA** [2], and **MMLU** [83] clinical topics. These datasets span various professional medical fields, research domains, and consumer health queries. By integrating these varied sources, MultiMedQA provides a robust platform for assessing the performance of language models in understanding and responding to medical inquiries (Fig. 6.4).

The deployment of Med-PaLM illustrates the transformative potential of LLMs in healthcare. These models are not only capable of understanding complex medical terminology and concepts but also of providing nuanced and precise answers. The journey of integrating these technologies into healthcare is just beginning, and the potential benefits are boundless.

6.3.3 Conclusion

As we conclude this exploration of Medical NLU and Conversation Applications, we marvel at the profound impact these technologies have had on healthcare. The integration of Natural Language Understanding (NLU) has ushered in an era of more responsive, personalized, and empathetic patient care.

From the pioneering efforts of ELIZA and PARRY to the sophisticated capabilities of IBM Watson and the practical applications of systems like Quro and MamaBot, we have witnessed the transformative power of conversational AI. These systems not only enhance the efficiency of medical consultations but also provide crucial emotional support, particularly for individuals managing chronic conditions or mental health challenges.

6.4 Legal Applications in NLU

In the hallowed halls of law, where the ink of statutes and the echoes of arguments weave the fabric of justice, a new force is emerging. NLU-embodied conversation agents transform the legal domain with their uncanny ability to comprehend and interact with human language. This chapter ventures into the heart of this revolution, where technology meets tradition and NLU's cutting-edge capabilities invigorate the age-old practice of law.

The legal field grapples with vast information, from case laws and precedents to contracts and client communications. In this complex landscape, NLU-powered conversation systems are emerging as indispensable allies, providing legal professionals with a potent tool to navigate the intricacies of their craft.

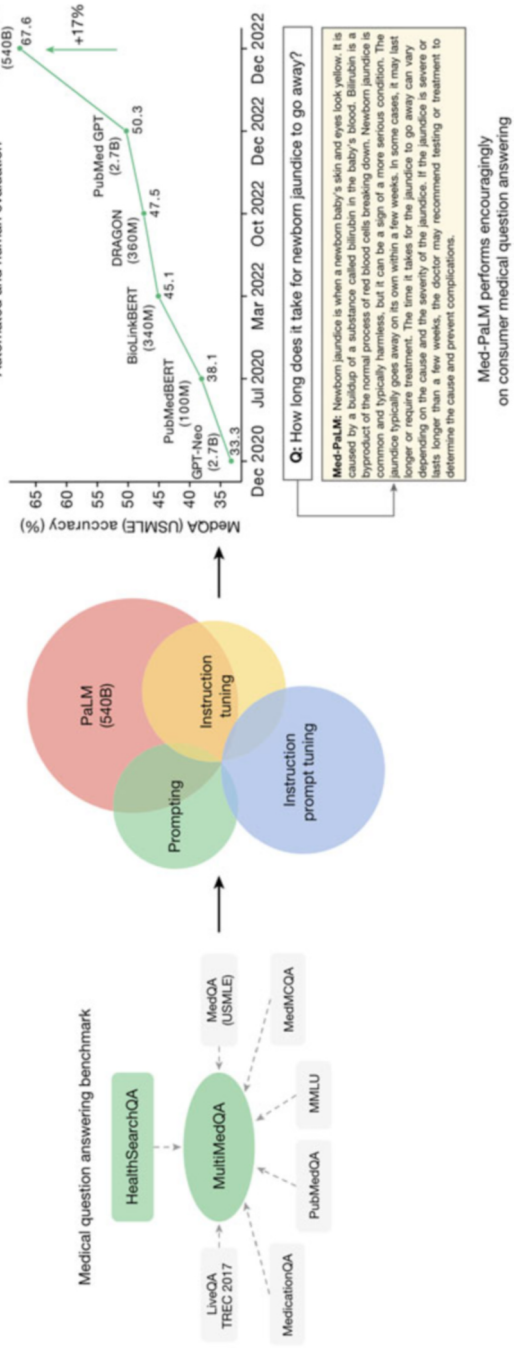

Fig. 6.4 Overview of MultiMedQA, a benchmark for answering medical questions spanning medical exam, medical research, and consumer medical questions [195]

Imagine a busy law office where attorneys are immersed in research and client consultations. Here, an NLU-enabled conversation agent operates as a virtual legal assistant, capable of answering client inquiries, scheduling appointments, and even providing preliminary legal advice. These chatbots, with their ability to understand and process natural language, bridge the gap between clients and their legal representatives, ensuring that essential communication is both efficient and accurate.

The reasoning ability of these legal domain conversation systems extends to legal research and document management as well. NLU-powered systems can swiftly analyze legal documents, identify relevant precedents, and suggest potential legal strategies. This capability significantly enhances the efficiency of legal research, enabling lawyers to craft more compelling and informed arguments.

In this section, we will explore case studies that illustrate the transformative impact of NLU-augmented chatbots in the legal domain. From automating routine tasks and enhancing client interactions to providing unparalleled access to legal information, these examples will highlight how the successful NLU is reshaping law practice, making it more responsive and accessible.

6.4.1 Legal Application Case Studies

TAXMAN: Early Legal Reasoning with NLU In the legal reasoning domain, the **TAXMAN** [138] project is a pioneering effort to leverage conversation agents to analyze legal cases. This can interpret and analyze the facts of corporate reorganization cases. By automating the interpretation of complex legal scenarios, TAXMAN provides a structured and methodical breakdown of cases, identifying key legal concepts and conclusions derived from the given facts.

The potential applications of TAXMAN extend beyond mere case analysis. By integrating NLU technologies, TAXMAN exemplifies how AI can assist in legal research, document review, and even the drafting of legal arguments. Its ability to process vast amounts of information and draw logical conclusions makes it an invaluable tool for legal practitioners navigating the complexities of corporate law. As we delve deeper into Legal NLU and Conversation Applications, TAXMAN is a testament to the transformative power of conversation agents in the legal sector.

LegalBot: A Deep Learning-Based Conversational Agent in the Legal Domain In the evolving legal technology landscape, **LegalBot** emerges as an LSTM encoder decoder-based conversational agent designed to assist with legal inquiries. LegalBot's generative nature allows it to craft responses dynamically, drawing on a vast repository of legal information and contextual understanding. This capability ensures that users receive not only accurate but also insightful responses tailored to the nuances of their specific queries. Automating the process of answering legal questions significantly enhances efficiency, reduces the workload on legal practitioners, and improves access to legal information for the general public.

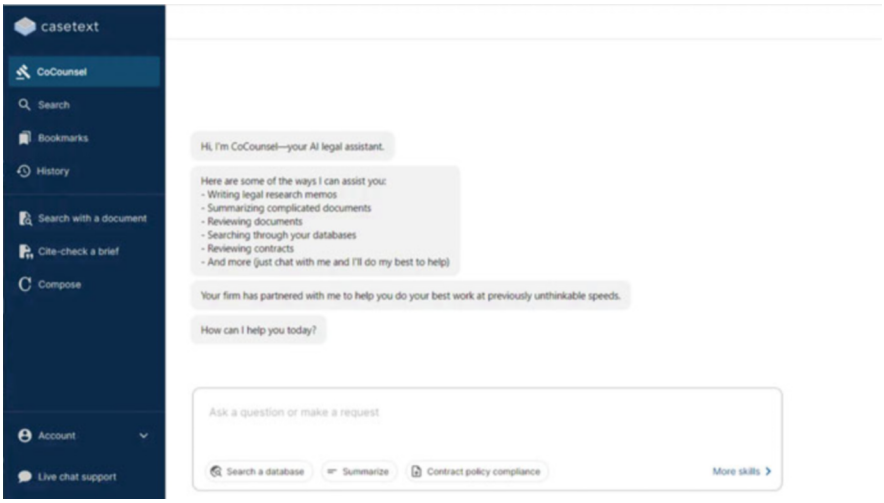

Fig. 6.5 The user interface of Application CoCounsel [8]

CoCounsel: The First AI Legal Assistant Made for Lawyers CoCounsel [25] emerges as a pioneering AI legal assistant specifically designed for lawyers. Built on the OpenAI's GPT-4, CoCounsel represents a significant milestone as the first legal LLM to enter the market and successfully pass the Bar Exam. CoCounsel's training encompasses a range of up-to-date legal materials, including case law, statutes, and other essential resources. This extensive knowledge base enables CoCounsel to deliver precise and informed responses to various legal queries, as shown in Fig. 6.5.

From conducting thorough document reviews and drafting detailed legal research memos to preparing for depositions and analyzing contracts, CoCounsel enhances the efficiency and effectiveness of legal practice. CoCounsel's ability to quickly and accurately process complex legal information showcases the transformative potential of NLU-augmented conversation systems in the legal domain. It not only streamlines routine tasks but also ensures that legal advice is grounded in the most current and relevant legal frameworks.

LeClari: Leveraging Event Schema for Clarifying Questions in Legal Case Retrieval Preliminary studies on using state-of-the-art large language models (LLMs) such as GPT-4 have revealed several challenges, including question duplication and the generation of low-utility content. To address these issues, the innovative system **LeClari** was developed, leveraging legal event schemas as external knowledge to instruct LLMs in producing more effective clarifying questions. LeClari has two core modules: a prompt module and a novel legal event selection module. The prompt module is designed to define prompts enriched with legal events to guide the generation of clarifying questions. Meanwhile, the legal event selection module meticulously selects potential event types by modelling the relationships among legal event types, conversational context, and candidate cases.

To further refine the system, LeClari employs ranking-oriented rewards and utilizes the reward-augmented maximum likelihood (RAML) method. This optimization strategy is directly based on the conversational legal search system's final retrieval performance, ensuring that the clarifying questions generated enhance the accuracy and efficiency of legal case retrieval.

6.4.2 Conclusion

As we conclude our exploration of Legal NLU and Conversation Applications, we reflect on the advancements these technologies have made in the legal field.

From the pioneering work of TAXMAN in legal reasoning to the conversational capabilities of LegalBot, we have seen how NLU can enhance the precision and efficiency of legal tasks. These systems not only streamline document review, legal research, and contract analysis but also provide insightful and contextually relevant responses to complex legal queries. CoCounsel stands out as a beacon of innovation, illustrating the power of large language models to assist lawyers in performing a variety of tasks with remarkable speed and accuracy. Its ability to process up-to-date legal materials and generate informed answers showcases the transformative potential of AI in legal practice. LeClari further exemplifies the innovative fusion of legal expertise and advanced AI technology. By leveraging legal event schemas to generate effective clarifying questions, LeClari significantly enhances the performance of legal conversational search systems, ensuring that legal professionals can retrieve the most relevant and precise information efficiently.

As we have explored these applications, it is evident that NLU technologies are augmenting human expertise and reshaping the very foundations of legal practice. They provide tools that enhance the accuracy, speed, and accessibility of legal services, ultimately contributing to a more efficient and just legal system.

Chapter 7
Challenges, Conclusion, and Future Direction

7.1 Challenges in Natural Language Understanding

As journeyed through the preceding chapters, we have marvelled at the significant advancements in Natural Language Understanding (NLU) brought about by deep learning. Yet, despite these strides, the path to truly comprehending human language in all its depth and diversity is still laden with formidable challenges.

Language, with its inherent complexities, presents a labyrinth of ambiguity, contextual nuances, and the vast expanse of human expression. It is like a living organism, continuously evolving, adapting, and presenting new facets to decode. Beyond these inherent intricacies, the real-world deployment of NLU models reveals even more hurdles. Issues of domain adaptation, the scarcity and variability of high-quality data, and ethical quandaries cast long shadows over the field.

This section delves into the heart of these challenges, exploring the obstacles that still impede our quest for robust and universally applicable NLU systems. It's a narrative of an ongoing struggle, a call for relentless innovation, and a reminder that the journey of understanding human language is as complex and dynamic as the language itself.

7.1.1 Ambiguity and Contextual Understanding

Language, in its essence, is a tapestry woven with threads of ambiguity and contextual richness. Each word and sentence carries the potential for multiple meanings, shifting and morphing depending on the surrounding context. This fluidity, while a testament to the beauty and complexity of human communication, presents a formidable challenge for Natural Language Understanding (NLU) models.

Imagine a single word, "bat." In one context, it flits through the night sky, a creature of darkness. In another, it rests in the hands of a cricketer, poised for a strike.

S. C. Han et al., *Natural Language Understanding in Conversational AI with Deep Learning*, https://doi.org/10.1007/978-3-031-74364-1_7

Such homonyms—words that sound alike but diverge in meaning—are just one of the puzzles NLU systems must solve. The challenge is compounded with polysemy, where a single word harbors multiple meanings, each waiting to be unlocked by the correct contextual key.

Consider the word "bank." To one person, it conjures images of a financial institution, the keeper of wealth. To another, it shows the serene image of a river's edge, where water gently kisses the land. The NLU model must discern from the surrounding text whether it is navigating the world of finance or nature.

This inherent ambiguity demands a level of contextual understanding that is both deep and nuanced. NLU models must not only recognize words but also grasp the intricate dance of context that gives these words their true meaning. They must learn to see beyond the surface, to the underlying intentions and connotations, much like a detective piecing together clues to solve a mystery.

In the human conversation, context is king. In a formal meeting, a phrase uttered in jest among friends can take on a starkly different tone. The subtleties of tone, the play of words, and the shared knowledge between speakers all contribute to the rich tapestry of meaning. Capturing this dynamic interplay in a model is akin to bottling lightning—an elusive and ongoing pursuit.

As we venture further into the depths of NLU, the challenge of ambiguity and contextual understanding is a testament to human language's complexity. It reminds us that, while we have made great strides, the journey toward fully grasping the nuances of our ever-evolving language continues.

7.1.2 Long-Term Dependencies

In the intricate dance of conversation, the ability to remember and build upon past interactions is paramount. Each exchange, each turn in dialogue, carries echoes of previous words and ideas, weaving a coherent narrative that makes sense of the unfolding interaction. Yet, for NLU models, capturing these long-term dependencies remains a daunting challenge, akin to tracing the delicate threads of a spider's web in a storm.

Consider a conversation over several minutes or even hours. Early statements, questions, or topics might resurface, requiring the model to recall and reference them accurately. This task is straightforward for humans, who effortlessly juggle multiple conversational threads, remembering past exchanges and anticipating future ones. However, for NLU systems, maintaining this continuity over extended interactions often proves elusive.

Current models, despite their sophistication, frequently stumble in this area. They struggle to hold onto the context as the conversation progresses, leading to responses that are disjointed or irrelevant. It's as if the model has a short memory, capable of responding to immediate prompts but unable to keep track of the conversation's overarching narrative.

Imagine a scenario where a user discusses holiday plans with a conversational agent. They might start by mentioning a preference for tropical destinations, later inquire about flight availability, and finally seek recommendations for local activities. An effective NLU system would seamlessly integrate these elements, recognizing that all these queries are part of a single, coherent topic. However, a system that fails to capture long-term dependencies might treat each query in isolation, offering generic responses that ignore the established context.

The root of this challenge lies in the temporal nature of language. Conversations are not static; they evolve, with each turn building upon the last. This requires models to develop a memory that is not just short-term but capable of spanning the entire dialogue. Despite advancements in techniques such as attention mechanisms and memory networks, the perfect solution remains elusive.

In multi-turn dialogues, where interactions are layered and complex, the need for models to grasp long-term dependencies becomes even more critical. The goal is to move beyond simple, transactional exchanges to create systems that can engage in meaningful, sustained conversations, mirroring the fluidity and depth of human interaction.

As we navigate the frontier of NLU, the challenge of capturing long-term dependencies stands as a testament to the complexity of human communication. It is a reminder that the path to true conversational intelligence is long and winding, demanding innovation, patience, and an ever-deepening understanding of the nuanced tapestry of dialogue.

7.1.3 Domain Adaptation

In the vast and varied landscape of language, context is everything. NLU models, meticulously trained on specific datasets, often falter when they venture into new and unfamiliar domains. It's as if these models, once comfortable in their native habitat, become disoriented in foreign terrain, struggling to comprehend the unfamiliar dialects and nuances they encounter.

Consider a model trained extensively on news articles. Within this domain, it navigates adeptly, parsing political analyses and economic reports with precision. However, when this same model is deployed to interpret medical literature or engage in casual conversation, it stumbles. The language, terminology, and contextual cues differ so markedly that the model's prior training becomes a hindrance rather than a help.

This challenge underscores the critical importance of transfer learning and domain adaptation techniques. These methods aim to equip NLU models with the flexibility to transition smoothly across different fields, retaining their efficacy and relevance. Yet, despite their promise, these techniques are still evolving, seeking the refinement needed to achieve robust cross-domain performance.

Imagine an NLU model as an explorer, venturing from the familiar shores of its training data into the uncharted waters of a new domain. For this journey, the

explorer must adapt and learn new cultural contexts and linguistic norms. Transfer learning provides the initial bridge, allowing the model to use general linguistic skills. Domain adaptation, on the other hand, fine-tunes these skills, aligning them with the specific characteristics of the new domain.

The process is akin to learning a new dialect within a language. While the basic structure and vocabulary remain the same, the subtleties—the idioms, the slang, the cultural references—require a period of immersion and adjustment. For NLU models, this immersion involves exposure to domain-specific data, enabling them to grasp the unique patterns and nuances that define each field.

Despite the progress made, the journey toward seamless domain adaptation is far from complete. Each domain presents its own set of challenges, demanding tailored solutions that go beyond mere surface-level adjustments. The goal is to develop models that are not just versatile but deeply attuned to the intricacies of each domain they encounter.

As we continue to push the boundaries of NLU, the quest for effective domain adaptation remains a central theme. It is a reminder of the rich diversity of human language and the complexity of creating systems that can navigate this diversity with grace and accuracy. The path forward is one of continuous learning and adaptation, mirroring the dynamic nature of language itself.

7.1.4 Data Scarcity and Quality

In the ever-evolving landscape of Natural Language Understanding, data stands as the cornerstone of progress. High-quality, annotated datasets are the essential bedrock upon which effective NLU models are constructed. However, creating these invaluable resources is a task of immense scale, demanding a significant investment of time, effort, and expertise. This laborious process of curation often poses a formidable obstacle to advancement, particularly when the aim is to develop models that are universally applicable.

Imagine the meticulous work required to annotate a dataset. Each sentence must be carefully labelled, and every nuance of meaning must be precisely captured. This labor-intensive endeavor necessitates not only linguistic acumen but also a profound understanding of the specific domain at hand. Creating such datasets is a slow, intricate dance, indispensable yet fraught with challenges.

The scarcity of diverse datasets further complicates this landscape. Our world is a rich tapestry of languages and dialects, each with its distinct structure, idioms, and cultural references. However, most available datasets are concentrated in a few dominant languages, leaving a significant portion of linguistic diversity unrepresented. This limitation hampers the development of NLU models that can seamlessly operate across different linguistic and cultural contexts.

Consider the plight of an NLU model trained exclusively on English texts. When tasked with understanding or generating text in a less-represented language, such as Swahili or Basque, it flounders. The nuances, idiomatic expressions, and cultural

references unique to these languages remain out of reach, a stark reminder of the gaps in our data resources. The absence of diverse, high-quality datasets thus restricts the horizons of NLU, confining its capabilities to well-trodden paths.

Moreover, the quality of available data is often inconsistent. While some datasets are meticulously annotated and rich in detail, others may be sparse or rife with errors. The reliability and accuracy of these datasets directly impact the performance of NLU models, underscoring the critical need for rigorous standards in data curation.

As we navigate the challenges of data scarcity and quality, the path forward is clear: we must invest in creating diverse, high-quality datasets that reflect the full spectrum of human language. This endeavor, though resource-intensive, is essential for the evolution of truly versatile and universally applicable NLU models. It is a journey of continuous discovery and refinement, mirroring the dynamic and multifaceted nature of language itself.

7.1.5 Bias and Fairness

In the intricate world of Natural Language Understanding, the quest for fairness and impartiality is a profound and ongoing challenge. As NLU models become increasingly sophisticated, they often inherit biases present in the data on which they are trained. These biases, subtle or overt, can manifest in ways that lead to unfair or skewed outcomes, casting a shadow over the promise of truly equitable AI systems.

Imagine a model trained on a dataset that reflects historical prejudices or stereotypes. If not carefully managed, these ingrained biases can ripple through the model's responses, subtly perpetuating inequalities rather than challenging them. This issue is not merely theoretical; it has real-world implications. A model might, for instance, exhibit gender or racial bias in its interactions, producing results that reinforce harmful stereotypes rather than promoting inclusivity.

Bias in NLU systems often begins with the data itself. Training datasets are shaped by the world they represent, and if that world is unequal or prejudiced, those same flaws can be mirrored in the AI's outputs. This is akin to a painter using imperfect pigments—no matter how skilled the quality of the materials will taint the artist, the final canvas.

Addressing this issue requires a meticulous approach to both data collection and model training. It begins with scrutinizing and diversifying the datasets used. Ensuring that data is representative of various demographics and perspectives is crucial. This means actively seeking out and incorporating voices and viewpoints that might be overlooked. It also involves applying rigorous checks to identify and rectify biases within the data.

Model training processes must also be refined to combat bias. Techniques such as fairness-aware algorithms and de-biasing strategies are emerging to address these concerns. However, these are not silver bullets; they require careful implementation

and continuous monitoring. Bias mitigation is a dynamic process, necessitating ongoing evaluation and adjustment to reflect evolving societal standards and values.

The challenge of ensuring fairness in NLU systems reflects broader societal issues, and as such, it demands a thoughtful and proactive approach. It is a call to action for developers and researchers to remain vigilant, question the sources of their data, and strive for systems that not only understand language but do so in a just and equitable manner.

In the end, the pursuit of fairness in NLU is a journey—one that requires constant vigilance, critical reflection, and a commitment to creating AI systems that serve all people with dignity and respect. It is an evolving challenge, mirroring the complexities of human society itself, and it remains at the forefront of the ongoing dialogue about the ethical use of technology.

7.1.6 Explainability and Interpretability

In Natural Language Understanding, deep learning models are frequently likened to enigmatic black boxes. These sophisticated algorithms, with their layers of neural networks and complex computations, often need to be made more clear of how decisions are made. This opacity can be a double-edged sword: while these models may achieve remarkable performance, their lack of transparency can be a significant hurdle, particularly in applications where trust and accountability are paramount.

Imagine a scenario where an NLU model recommends or delivers a decision in a critical context, such as a medical diagnosis or legal judgment. The stakes are high, and the need for clarity becomes urgent. Yet, when confronted with the model's output, stakeholders may grapple with uncertainty. How did the model arrive at its conclusion? What factors influenced its decision? Without a clear understanding of the model's reasoning process, trusting and validating its outcomes becomes a challenging endeavor.

The black-box nature of deep learning models stems from their intricate architectures and the vast number of parameters they use. These models learn to represent complex patterns in data through numerous hidden layers, but this process can make it difficult to trace how a specific input leads to a particular output. This lack of explainability can be particularly problematic in sensitive applications where decisions have significant consequences for individuals or society.

Addressing this challenge requires a concerted effort to enhance explainability and interpretability. Techniques in this domain aim to shed light on the decision-making processes of these models, providing insights into how inputs are transformed into outputs. Methods such as attention mechanisms, saliency maps, and model-agnostic explanations seek to bridge the gap between the model's internal workings and human understanding.

Imagine these techniques as tools that offer glimpses into the model's thought process. Attention mechanisms, for instance, can highlight which parts of the input data were most influential in generating a specific response, offering a clearer picture

of the model's focus. Saliency maps can illustrate which features contributed most to the decision, providing a visual representation of the model's reasoning.

Nevertheless, achieving true interpretability remains a complex and ongoing pursuit. The challenge lies not only in developing methods that provide meaningful insights but also in ensuring that these methods are accessible and useful to those who need them. This is especially critical in domains where the consequences of decisions are profound and far-reaching.

Essentially, the quest for explainability and interpretability in NLU is about making the invisible visible. It is about transforming the black box into a transparent vessel that can be understood, trusted, and held accountable. As we advance in our ability to demystify these models, we move closer to creating systems that not only excel in performance but also foster trust and confidence in their applications.

7.2 Conclusion

As we reflect on the journey of single-turn and multi-turn Natural Language Understanding (NLU) within deep learning-based conversational AI, it is evident that the field has traversed impressive milestones. The evolution of deep learning has ushered in a new era of linguistic capabilities, marked by the advent of sophisticated models like transformers and attention mechanisms. These innovations have dramatically enhanced our ability to interpret and generate human language with unprecedented accuracy and contextual awareness.

From the simplicity of single-turn exchanges to the complexity of multi-turn dialogues, NLU systems have shown remarkable adaptability. They now handle a broad spectrum of conversational scenarios, moving from transactional interactions to intricate, nuanced conversations that mirror the subtleties of human dialogue. This progress is not just a testament to technological prowess but also a reflection of the increasing sophistication in evaluating and refining these systems. Effective evaluation mechanisms have been crucial in pushing the boundaries, ensuring that NLU systems continuously evolve to meet new challenges and expectations.

However, as we navigate these advancements, it becomes clear that the road ahead is paved with significant challenges. Language, with its inherent ambiguity and complexity, continues to test the limits of our models. The task of maintaining coherence over extended conversations, adapting to diverse domains, and ensuring the quality and fairness of data remains a formidable one. Issues of bias, the need for interpretability, and the demands on resources highlight the areas where further research and innovation are beneficial and essential.

Resolving these challenges is a call to action for the research community, developers, and stakeholders alike. It requires a concerted effort to delve deeper into the intricacies of language and build models that are not only powerful but also equitable and transparent. As we move forward, the focus must remain on advancing both the technological capabilities and the ethical standards that govern the use of NLU systems.

In essence, while the strides made in NLU are impressive, they serve as a stepping stone toward a more nuanced and comprehensive understanding of human language. The journey is ongoing, characterized by a dynamic interplay of progress and challenges. It is a reminder that the quest for truly intelligent, fair, and context-aware conversational systems is a continuous endeavor—one that holds the promise of further innovation and insight as we strive to bridge the gap between human and machine understanding.

7.3 Future Direction

As we stand on the precipice of a new era in Natural Language Understanding, the horizon brims with untapped potential and intriguing possibilities. The journey thus far has witnessed remarkable strides, propelled by the relentless march of deep learning and the burgeoning capabilities of conversational AI. Yet, the road ahead is as promising as it is challenging, demanding not only ingenuity but also a nuanced understanding of the evolving landscape.

The future of NLU is poised to unravel new dimensions, each presenting its own set of opportunities and hurdles. From refining our grasp of intricate contextual cues to crafting models that transcend linguistic and cultural boundaries, the forthcoming innovations are set to redefine the boundaries of what conversational AI can achieve. In this unfolding narrative, each chapter holds the promise of addressing the current limitations and exploring novel avenues for advancement.

The quest for improvement will be characterized by a deeper engagement with the subtleties of human language and a more sophisticated interplay between technology and human oversight. As we venture into this new realm, the focus will shift toward enhancing contextual understanding, overcoming biases, and making AI systems more transparent and accessible. This journey is not merely about technological enhancement but also about fostering a more harmonious integration of AI into our daily lives.

In the pages that follow, we will delve into the key areas poised for development, exploring how these future directions could shape the next generation of NLU systems and, by extension, transform the way we interact with technology. The promise of these advancements invites us to envision a future where conversational AI not only understands our language but also resonates with our diverse and dynamic human experience.

7.3.1 Improved Contextual Understanding

In conversational AI, the quest for a deeper and more nuanced understanding of context is nothing short of a grand pursuit. The richness of human dialogue lies in its intricate layers of meaning, often shaped by preceding exchanges and the subtleties

of personal experiences. For AI to truly grasp the essence of these conversations, it must go beyond surface-level interactions and delve into the depths of long-term contextual continuity.

At present, conversational models excel at parsing and responding to immediate queries with impressive accuracy. However, their ability to maintain and utilize context over extended dialogues remains challenging. This limitation often results in disjointed or irrelevant responses, detracting from the fluidity of human-like interaction. The future of NLU will hinge on developing more sophisticated mechanisms that can emulate the intricate dance of human memory and understanding.

Imagine an AI system that, like a seasoned conversationalist, recalls the details of past interactions with the same ease and relevance as a human counterpart. Achieving this vision will necessitate advances in memory mechanisms—innovations that allow models to not only store but also meaningfully access and apply contextual information from earlier conversations. These mechanisms will need to be dynamic, adapting to the flow of dialogue and the evolving needs of the conversation.

Moreover, integrating external knowledge bases holds immense promise in enhancing contextual understanding. By drawing upon a vast reservoir of information beyond the immediate conversation, AI can enrich its responses with deeper insights and more relevant content. This involves not just accessing data but weaving it seamlessly into the dialogue, providing responses that are both informed and contextually appropriate.

In this emerging landscape, the interplay between internal memory systems and external knowledge sources will be pivotal. Models that effectively combine these elements will not only offer more coherent and contextually aware interactions but also bridge the gap between AI and the nuanced subtleties of human conversation.

As we forge ahead, we will focus on refining these mechanisms and ensuring they contribute to a more authentic and engaging conversational experience. The goal is not merely to enhance AI's technical capabilities but to cultivate a conversational partner who resonates with the depth and complexity of human communication. This pursuit will define the next frontier in NLU, where context is not just understood but seamlessly integrated into the very fabric of interaction.

7.3.2 Advanced Transfer Learning Techniques

As we navigate the intricate landscape of Natural Language Understanding, the pursuit of versatility and adaptability in conversational AI emerges as a defining challenge. One of the most promising avenues for achieving this is advanced transfer learning techniques, which promise to revolutionize how models adapt to new domains with minimal data. This capability is crucial for enhancing the flexibility and applicability of NLU systems across diverse contexts and applications.

Transfer learning, in its essence, involves leveraging knowledge gained from one domain to inform and improve performance in another. Traditional approaches often rely on substantial datasets from target domains to fine-tune models effectively.

However, in many real-world scenarios, acquiring such data can be impractical or infeasible. Enter advanced techniques like meta-learning and zero-shot learning—innovations poised to transform this paradigm.

Meta-learning, or "learning to learn," represents a sophisticated evolution of transfer learning. By training models on a diverse array of tasks, meta-learning equips them with the ability to generalize from limited examples. This approach allows models to rapidly adapt to new, unseen tasks by drawing on their prior experiences. The essence of meta-learning lies in its ability to streamline the adaptation process, making it possible for NLU systems to excel in new domains with minimal additional data. Imagine a conversational agent that, after being exposed to a broad spectrum of topics, can seamlessly switch to addressing niche subjects easily and accurately.

Zero-shot learning, on the other hand, pushes the boundaries of transfer learning by enabling models to handle tasks or concepts they have never encountered during training. This technique leverages semantic representations and relationships to infer new classes or domains. For instance, a model trained on general conversational data can utilize zero-shot learning to engage in meaningful dialogue about a specialized subject, such as legal or medical terminology, without having been explicitly trained on that data.

The synergy between meta-learning and zero-shot learning presents a powerful combination for advancing NLU. Meta-learning provides the foundational capability to adapt swiftly to new scenarios, while zero-shot learning expands the model's reach into uncharted territories. Together, these techniques enhance the model's ability to handle various tasks and domains, increasing its utility and applicability.

As we look to the future, the development and refinement of these advanced transfer learning techniques will be pivotal in creating NLU systems that are not only more adaptable but also more intelligent in their application. The ability to effectively generalize from minimal data and tackle novel challenges will mark a significant leap forward in the quest for robust conversational AI.

7.3.3 Multilingual and Cross-cultural NLU

In an increasingly interconnected world, the ability of Natural Language Understanding systems to navigate multiple languages and diverse cultural contexts is emerging as a critical frontier. As conversational AI seeks to transcend geographical and linguistic boundaries, the quest for multilingual and cross-cultural competence becomes not merely advantageous but essential. This journey toward global applicability requires more than technical innovation—it demands a deep understanding of linguistic diversity and cultural nuances.

Imagine a world where conversational AI seamlessly bridges the gaps between languages, facilitating communication across continents with the fluidity of a native speaker. Such a vision is within reach, but it hinges on two pivotal aspects: the creation of expansive, diverse datasets and the development of models capable of generalizing across languages.

To achieve true multilingual capability, it is imperative to construct datasets that are not only vast but also representative of a wide array of languages and dialects. The richness of these datasets must capture the subtleties of various linguistic structures, idioms, and contextual expressions. This means going beyond the commonly spoken languages to include underrepresented and lesser-known tongues, ensuring that AI systems are equipped to understand and generate language with genuine proficiency across the globe.

Moreover, the challenge extends beyond mere translation; it encompasses the ability to interpret and respect cultural nuances. Language is deeply intertwined with culture, and effective communication involves recognizing and adapting to these cultural contexts. A model trained solely on English data, for instance, might struggle with the cultural references and social norms embedded in other languages. Thus, developing AI systems that can comprehend and respond appropriately in a cross-cultural setting is crucial for fostering truly global interactions.

To address these challenges, the field is turning to advanced methodologies in transfer learning and cross-lingual embeddings. Techniques such as multilingual BERT and cross-lingual language models aim to create shared representations that enable a deeper understanding across different languages. These models are trained to grasp the commonalities between languages while appreciating their unique characteristics. The result is a more cohesive and adaptable NLU system capable of switching seamlessly between languages and cultural contexts.

In addition, ongoing research into zero-shot learning and few-shot learning techniques is helping models perform well in languages with limited training data. By leveraging knowledge from well-resourced languages, these techniques enable AI systems to infer and generate text in less commonly spoken languages with remarkable accuracy.

The path to multilingual and cross-cultural NLU is a journey of both technical and cultural exploration. As we advance, the focus will be on refining models to handle linguistic diversity and honor the richness of human culture. In doing so, conversational AI will become more versatile and inclusive, paving the way for truly global dialogue and understanding.

7.3.4 Ethics and Fairness

The quest for technological excellence is paralleled by an equally pressing imperative: ensuring ethical integrity and fairness. As conversational AI systems become increasingly embedded in our daily lives, the importance of addressing biases and fostering equitable outcomes cannot be overstated. The journey toward ethical AI is not merely a technical endeavor but a profound commitment to building systems that reflect our collective values of fairness and inclusivity.

Imagine a world where conversational AI interacts with individuals from diverse backgrounds without perpetuating existing societal biases. This vision necessitates a concerted effort to scrutinize, measure, and mitigate biases that may lurk within

these systems. The challenge lies not only in recognizing the subtle ways in which bias can manifest but also in devising methods to counteract its influence.

To navigate this complex terrain, researchers and practitioners must establish robust frameworks for evaluating fairness. This involves developing comprehensive metrics that can capture a range of potential biases, from gender and racial biases to socioeconomic and linguistic disparities. Traditional evaluation methods often need to catch up, as they may overlook nuances or need to account for context-specific factors. Therefore, there is a pressing need for innovative approaches that offer a more nuanced understanding of fairness and bias in AI systems.

Moreover, implementing techniques to ensure unbiased model behavior is equally crucial. This includes designing algorithms resilient to bias and incorporating fairness-aware mechanisms during the training and deployment phases. Techniques such as adversarial debiasing, where models are trained to counteract bias by exposing them to challenging examples, and fairness constraints, which adjust the model's decision-making process to uphold ethical standards, are gaining traction. These methods aim to create AI systems that not only perform well but also uphold the principles of fairness and equality.

Furthermore, transparency and accountability play a pivotal role in the ethical deployment of NLU systems. Ensuring that models are interpretable and that their decision-making processes can be scrutinized is essential for building trust and addressing concerns related to fairness. By fostering an environment where AI systems are open to review and critique, developers can better identify and rectify biases and ensure that their models serve all users equitably.

As we forge ahead, integrating ethical considerations into the development and deployment of NLU systems will be instrumental in shaping a future where AI contributes positively to society. This journey is not merely about technological advancement but about aligning our innovations with the fundamental values of justice and fairness. By embracing these principles, we can aspire to create conversational AI that not only understands and interacts with us but does so in a manner that is just, inclusive, and reflective of our shared human dignity.

7.3.5 Resource-Efficient Models

In the rapidly evolving landscape of Natural Language Understanding, the quest for cutting-edge performance often clashes with the practical constraints of computational resources. As AI systems become more sophisticated, they frequently demand increasingly substantial amounts of data, processing power, and memory. This presents a paradox: while we strive for models that excel in accuracy and capability, we must also address the need for efficiency and accessibility. The future of NLU lies in striking a balance between high performance and resource efficiency, making advanced technologies available to a broader audience.

Imagine a world where state-of-the-art conversational AI can be deployed not only by tech giants with vast computational resources but also by smaller enter-

prises and individuals with limited means. This vision is attainable by developing resource-efficient models—systems designed to deliver exceptional performance while minimizing their computational footprint.

One key technique in achieving this balance is model pruning. This process systematically removes parts of a neural network that contribute little to its overall performance. By trimming away these redundant elements, we can create a leaner, faster model that retains its core capabilities. Pruning not only reduces the computational load but also lessens the memory requirements, making it possible to deploy sophisticated NLU systems on less powerful hardware.

Another crucial method is quantization, which involves converting a model's weights and activations from high-precision floating-point numbers to lower-precision formats. This transformation significantly reduces the model's memory footprint and accelerates computation. Despite the reduction in precision, carefully applying quantization techniques can maintain a model's performance at levels comparable to its more resource-intensive counterparts.

The evolution of efficient architectures is also pivotal in this realm. Advances in neural network design continually push the boundaries of what can be achieved with fewer resources. Architectures such as MobileNets and EfficientNet are specifically crafted to balance performance with efficiency, enabling high-quality NLU while keeping computational demands in check. These architectures leverage novel strategies like depthwise separable convolutions and compound scaling to optimize resource usage without compromising accuracy.

As we look to the future, the emphasis on resource-efficient models will be instrumental in democratizing access to advanced NLU technologies. By making high-performance AI systems more accessible, we open doors for innovation across diverse sectors, from education and healthcare to small businesses and local communities. The drive toward efficiency not only enhances the feasibility of deploying AI solutions in resource-constrained environments but also aligns with broader goals of sustainability and inclusivity.

In this emerging landscape, the challenge is to blend technical prowess with pragmatic considerations, ensuring that our advancements in NLU serve a wider array of users and applications. Through continued innovation in model pruning, quantization, and efficient architectures, we can realize a future where advanced conversational AI is powerful and universally accessible, paving the way for a more equitable and technologically empowered world.

7.3.6 Human-AI Collaboration

As conversational AI systems become increasingly sophisticated, the nature of their interaction with human users is undergoing a profound transformation. The future of Natural Language Understanding is not merely about advancing technology but about reimagining the synergy between humans and machines. The goal is to cultivate a collaborative relationship where AI serves not as an autonomous

decision-maker but as a supportive assistant that enhances human capabilities and decision-making processes.

Picture a scenario where an AI system seamlessly integrates into the daily workflow, not as an isolated entity but as an active participant in a collaborative partnership. This vision highlights the importance of developing interfaces that facilitate a dynamic interaction between humans and AI, ensuring that the technology complements rather than competes with human expertise.

One crucial aspect of enhancing this interaction is the design of intuitive interfaces that promote clear and effective communication between users and AI systems. Such interfaces should be crafted to support natural, fluid exchanges, allowing users to engage with AI in ways that align with their cognitive processes and preferences. This might involve incorporating user-friendly dialogue boxes, visual aids, and interactive elements that make the AI's capabilities and limitations transparent and accessible.

Human oversight is another vital component in this collaborative framework. Ensuring that AI systems operate within boundaries set by human judgment helps prevent overreliance and mitigates potential risks associated with autonomous decision-making. By embedding mechanisms for oversight, such as review processes and decision checkpoints, users can maintain control and intervene when necessary. This approach not only enhances the reliability of AI systems but also builds trust by providing users with the assurance that they remain at the helm of critical decisions.

Moreover, adaptive feedback mechanisms are essential for refining the collaboration between humans and AI. These mechanisms allow AI systems to learn from user interactions, continuously improving their performance and responsiveness based on real-world usage. By incorporating feedback loops that capture user input and preferences, AI can evolve in a manner that better aligns with human needs and expectations.

In this collaborative paradigm, AI transitions from a mere tool to a proactive partner. Rather than operating in isolation, AI systems become integral to the human decision-making process, offering insights, suggestions, and support that enhance overall effectiveness. This shift not only optimizes AI's utility but also fosters a more harmonious and productive interaction between technology and its users.

As we advance, the focus will be on refining these collaborative interfaces and oversight mechanisms, ensuring that AI systems are not only advanced in their capabilities but also adept at working alongside humans. The future of NLU lies in creating a partnership where AI amplifies human strengths and compensates for limitations, paving the way for more effective, ethical, and empowering applications of conversational technology.

References

1. Abacha, A.B., Agichtein, E., Pinter, Y., Demner-Fushman, D.: Overview of the medical question answering task at trec 2017 liveqa. In: TREC, pp. 1–12 (2017)
2. Abacha, A.B., Mrabet, Y., Sharp, M., Goodwin, T.R., Shooshan, S.E., Demner-Fushman, D.: Bridging the gap between consumers' medication questions and trusted answers. In: MedInfo, pp. 25–29 (2019)
3. Abro, W.A., Aicher, A., Rach, N., Ultes, S., Minker, W., Qi, G.: Natural language understanding for argumentative dialogue systems in the opinion building domain. Knowl. Based Syst. **242**, 108318 (2022)
4. Abro, W.A., Qi, G., Aamir, M., Ali, Z.: Joint intent detection and slot filling using weighted finite state transducer and bert. Appl. Intell. **52**(15), 17356–17370 (2022)
5. Abu-Salih, B., Alotaibi, S., Abukhurma, R., Almiani, M., Aljaafari, M.: Dao-lgbm: dual annealing optimization with light gradient boosting machine for advocates prediction in online customer engagement. Cluster Comput., 1–27 (2024)
6. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
7. Al-Ajmi, A.H., Al-Twairesh, N.: Building an arabic flight booking dialogue system using a hybrid rule-based and data driven approach. IEEE Access **9**, 7043–7053 (2021)
8. Ambrogi, B.: Casetext Launches Co-Counsel, Its OpenAI-Based 'Legal Assistant' To Help Lawyers Search Data, Review Documents, Draft Memos, Analyze Contracts and More — lawnext.com. https://www.lawnext.com/2023/03/casetext-launches-co-counsel-its-openai-based-legal-assistant-to-help-lawyers-search-data-review-documents-draft-memos-analyze-contracts-and-more.html (2023). [Accessed 24-06-2024]
9. Angele, K., Angele, J.: Derool: A rule-based dialog engine. In: RuleML+ RR (Supplement), pp. 157–168 (2020)
10. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7), e0130140 (2015)
11. Bapna, A., Tur, G., Hakkani-Tur, D., Heck, L.: Sequential Dialogue Context Modeling for Spoken Language Understanding. In: Proceedings of the 18th SIGDIAL, pp. 103–114 (2017)
12. Baum, L.E., Petrie, T.: Statistical inference for probabilistic functions of finite state markov chains. Ann. Math. Stat. **37**(6), 1554–1563 (1966). http://www.jstor.org/stable/2238772
13. Béchet, F., Raymond, C.: Is ATIS too shallow to go deeper for benchmarking Spoken Language Understanding models? In: Interspeech 2018, pp. 1–5. ISCA, Hyderabad, India (2018). https://hal.inria.fr/hal-01835425

S. C. Han et al., *Natural Language Understanding in Conversational AI with Deep Learning*, https://doi.org/10.1007/978-3-031-74364-1

14. Bellomaria, V., Castellucci, G., Favalli, A., Romagnoli, R.: Almawave-slu: A new dataset for SLU in italian. In: Proceedings of the Sixth Italian Conference on Computational Linguistics. AILC, Bari, Italy (2019)
15. Bhargava, A., Celikyilmaz, A., Hakkani-Tür, D., Sarikaya, R.: Easy contextual intent prediction and slot detection. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 8337–8341. IEEE, Vancouver, Canada (2013)
16. Bhasin, A., Natarajan, B., Mathur, G., Jeon, J.H., Kim, J.S.: Unified parallel intent and slot prediction with cross fusion and slot masking. In: E. Métais, F. Meziane, S. Vadera, V. Sugumaran, M. Saraee (eds.) Natural Language Processing and Information Systems, pp. 277–285. Springer, Cham (2019)
17. Bhasin, A., Natarajan, B., Mathur, G., Mangla, H.: Parallel intent and slot prediction using mlb fusion. In: 2020 IEEE 14th International Conference on Semantic Computing (ICSC), pp. 217–220. IEEE (2020)
18. Bodigutla, P.K., Tiwari, A., Matsoukas, S., Valls-Vargas, J., Polymenakos, L.: Joint turn and dialogue level user satisfaction estimation on multi-domain conversations. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 3897–3909 (2020)
19. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
20. Brixey, J., Hoegen, R., Lan, W., Rusow, J., Singla, K., Yin, X., Artstein, R., Leuski, A.: Shihbot: A facebook chatbot for sexual health information on hiv/aids. In: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, pp. 370–373 (2017)
21. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Adv. Neural Inf. Process. Syst. **33**, 1877–1901 (2020)
22. Budzianowski, P., Wen, T.H., Tseng, B.H., Casanueva, I., Ultes, S., Ramadan, O., Gašić, M.: MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In: E. Riloff, D. Chiang, J. Hockenmaier, J. Tsujii (eds.) Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 5016–5026. Association for Computational Linguistics, Brussels, Belgium (2018). https://doi.org/10.18653/v1/D18-1547. https://aclanthology.org/D18-1547
23. Burgan, D.: Dialogue Systems and Dialogue Management. DST Group Edinburgh, Edinburgh SA, Australia (2016)
24. Casanueva, I., Temčinas, T., Gerz, D., Henderson, M., Vulić, I.: Efficient intent detection with dual sentence encoders. In: Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, pp. 38–45 (2020)
25. Casetext: CoCounsel | The First AI Legal Assistant, Made for Lawyers — casetext.com. https://casetext.com/cocounsel/ (2023). [Accessed 24-06-2024]
26. Castellucci, G., Bellomaria, V., Favalli, A., Romagnoli, R.: Multi-lingual intent detection and slot filling in a joint bert-based model (2019). arXiv:1907.02884
27. Castle-Green, T., Reeves, S., Fischer, J.E., Koleva, B.: Decision trees as sociotechnical objects in chatbot design. In: Proceedings of the 2nd Conference on Conversational User Interfaces, pp. 1–3 (2020)
28. Celikyilmaz, A., Hakkani-Tür, D.: A joint model for discovery of aspects in utterances. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 330–338. Association for Computational Linguistics, Jeju Island, Korea (2012). https://www.aclweb.org/anthology/P12-1035
29. Chaudhuri, D., Kristiadi, A., Lehmann, J., Fischer, A.: Improving response selection in multi-turn dialogue systems by incorporating domain knowledge. In: Proceedings of the 22nd Conference on Computational Natural Language Learning, pp. 497–507 (2018)
30. Chen, D., Huang, Z., Wu, X., Ge, S., Zou, Y.: Towards joint intent detection and slot filling via higher-order attention. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22. International Joint Conferences on Artificial Intelligence Organization, Vienna, Austria (2022)
31. Chen, L., Zhang, D., Mark, L.: Understanding user intent in community question answering. In: Proceedings of the 21st International Conference on World Wide Web, pp. 823–828 (2012)

32. Chen, M., Zeng, J., Lou, J.: A self-attention joint model for spoken language understanding in situational dialog applications (2019). arXiv:1905.11393
33. Chen, Q., Zhuo, Z., Wang, W.: BERT for joint intent classification and slot filling (2019). http://arxiv.org/abs/1902.10909
34. Chen, Y.N., Hakanni-Tür, D., Tür, G., Celikyilmaz, A., Guo, J., Deng, L.: Syntax or semantics? knowledge-guided joint semantic frame parsing. In: 2016 IEEE Spoken Language Technology Workshop (SLT), pp. 348–355. IEEE, San Diego, USA (2016)
35. Cheng, L., Jia, W., Yang, W.: An effective non-autoregressive model for spoken language understanding. In: CIKM '21: Proceedings of the 30th ACM International Conference on Information and Knowledge Management, pp. 241–250. Association for Computing Machinery, New York, USA (2021)
36. Chiba, Y., Ito, A., Ito, M.: Modeling user's state during dialog turn using hmm for multi-modal spoken dialog system. In: L. Miller, A. Culen (eds.) ACHI 2014 - 7th International Conference on Advances in Computer-Human Interactions, ACHI 2014 - 7th International Conference on Advances in Computer-Human Interactions, pp. 343–346. International Academy, Research and Industry Association, IARIA (2014). Publisher Copyright: Copyright © IARIA, 2014.; 7th International Conference on Advances in Computer-Human Interactions, ACHI 2014; Conference date: 23-03-2014 Through 27-03-2014
37. Cho, K., van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder–decoder approaches. In: D. Wu, M. Carpuat, X. Carreras, E.M. Vecchi (eds.) Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pp. 103–111. Association for Computational Linguistics, Doha, Qatar (2014). https://doi.org/10.3115/v1/W14-4012. https://aclanthology.org/W14-4012
38. Colby, K.M., Weber, S., Hilf, F.D.: Artificial paranoia. Artif. Intell. **2**, 1–25 (1971)
39. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
40. Costello, C., Lin, R., Mruthyunjaya, V., Bolla, B., Jankowski, C.: Multi-layer ensembling techniques for multilingual intent classification (2018). arXiv:1806.07914
41. Coucke, A., Saade, A., Ball, A., Bluche, T., Caulier, A., Leroy, D., Doumouro, C., Gisselbrecht, T., Caltagirone, F., Lavril, T., Primet, M., Dureau, J.: Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces (2018). arXiv:1805.10190
42. Cox, D.R.: The regression analysis of binary sequences. J. Roy. Stat. Soc. B (Methodol.) **20**(2), 215–232 (1958)
43. Daha, F., Hewavitharana, S.: Deep neural architecture with character embedding for semantic frame detection. In: 2019 IEEE 13th International Conference on Semantic Computing (ICSC), pp. 302–307. IEEE, Newport Beach, USA (2019)
44. Dahl, D.A., Bates, M., Brown, M., Fisher, W., Hunicke-Smith, K., Pallett, D., Pao, C., Rudnicky, A., Shriberg, E.: Expanding the scope of the ATIS task: The ATIS-3 corpus. In: Proceedings of the Workshop on Human Language Technology, pp. 43–48. Association for Computational Linguistics, Plainsboro, USA (1994). https://www.aclweb.org/anthology/H94-1010
45. Dai, Y., Zhang, Y., Ou, Z., Wang, Y., Feng, J.: Elastic crfs for open-ontology slot filling. Appl. Sci. **11**(22) (2021). https://doi.org/10.3390/app112210675. https://www.mdpi.com/2076-3417/11/22/10675
46. Deng, L., Tur, G., He, X., Hakkani-Tür, D.: Use of kernel deep convex networks and end-to-end learning for spoken language understanding. In: 2012 IEEE Spoken Language Technology Workshop (SLT), pp. 210–215. IEEE, Miami, USA (2012)
47. Deoras, A., Sarikaya, R.: Deep belief network based semantic taggers for spoken language understanding. In: Interspeech, pp. 2713–2717. ISCA, Lyon, France (2013)
48. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: J. Burstein, C. Doran, T. Solorio (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1423. https://aclanthology.org/N19-1423

49. D'Haro, L.F., Yoshino, K., Hori, C., Marks, T.K., Polymenakos, L., Kummerfeld, J.K., Galley, M., Gao, X.: Overview of the seventh dialog system technology challenge: Dstc7. Comput. Speech Lang. **62**, 101068 (2020)
50. Dinan, E., Logacheva, V., Malykh, V., Miller, A., Shuster, K., Urbanek, J., Kiela, D., Szlam, A., Serban, I., Lowe, R., Prabhumoye, S., Black, A.W., Rudnicky, A., Williams, J., Pineau, J., Burtsev, M., Weston, J.: The second conversational intelligence challenge (convai2). In: S. Escalera, R. Herbrich (eds.) The NeurIPS '18 Competition, pp. 187–208. Springer International Publishing, Cham (2020)
51. Du, M., Manjunatha, V., Jain, R., Deshpande, R., Dernoncourt, F., Gu, J., Sun, T., Hu, X.: Towards interpreting and mitigating shortcut learning behavior of nlu models. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 915–929 (2021)
52. E, H., Niu, P., Chen, Z., Song, M.: A novel bi-directional interrelated model for joint intent detection and slot filling. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5467–5471. Association for Computational Linguistics, Florence, Italy (2019). https://doi.org/10.18653/v1/P19-1544. https://www.aclweb.org/anthology/P19-1544
53. Elman, J.L.: Finding structure in time. Cognit. Sci. **14**(2), 179–211 (1990)
54. Eric, M., Goel, R., Paul, S., Sethi, A., Agarwal, S., Gao, S., Kumar, A., Goyal, A., Ku, P., Hakkani-Tur, D.: Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In: Proceedings of the Twelfth Language Resources and Evaluation Conference, pp. 422–428 (2020)
55. Ferrucci, D.A.: Introduction to "this is watson". IBM J. Res. Dev. **56**(3.4), 1:1–1:15 (2012). https://doi.org/10.1147/JRD.2012.2184356
56. Firdaus, M., Bhatnagar, S., Ekbal, A., Bhattacharyya, P.: A deep learning based multi-task ensemble model for intent detection and slot filling in spoken language understanding. In: L. Cheng, A.C.S. Leung, S. Ozawa (eds.) Neural Information Processing, pp. 647–658. Springer, Cham (2018)
57. Firdaus, M., Bhatnagar, S., Ekbal, A., Bhattacharyya, P.: Intent detection for spoken language understanding using a deep ensemble model. In: PRICAI 2018: trends in artificial intelligence: 15th pacific rim international conference on artificial intelligence, nanjing, china, august 28–31, 2018, proceedings, Part I 15, pp. 629–642. Springer (2018)
58. Firdaus, M., Bhatnagar, S., Ekbal, A., Bhattacharyya, P.: Intent detection for spoken language understanding using a deep ensemble model. In: PRICAI 2018: Trends in Artificial Intelligence: 15th Pacific Rim International Conference on Artificial Intelligence, Nanjing, China, August 28–31, 2018, Proceedings, Part I 15, pp. 629–642. Springer (2018)
59. Firdaus, M., Golchha, H., Ekbal, A., Bhattacharyya, P.: A deep multi-task model for dialogue act classification, intent detection and slot filling. Cognit. Comput. **13**, 626–645 (2020)
60. Firdaus, M., Kumar, A., Ekbal, A., Bhattacharyya, P.: A multi-task hierarchical approach for intent detection and slot filling. Knowl. Based Syst. **183**, 104846 (2019)
61. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat., 1189–1232 (2001)
62. Fu, T., Zhao, X., Yan, R.: Delving into global dialogue structures: Structure planning augmented response selection for multi-turn conversations. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 495–505 (2023)
63. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: International Conference on Machine Learning, pp. 1243–1252. PMLR (2017)
64. Ghosh, S., Bhatia, S., Bhatia, A.: Quro: facilitating user symptom check using a personalised chatbot-oriented dialogue system. Stud. Health Technol. Inform. **252**, 51–56 (2018)
65. Gong, Y., Luo, X., Zhu, Y., Ou, W., Li, Z., Zhu, M., Zhu, K.Q., Duan, L., Chen, X.: Deep cascade multi-task learning for slot filling in online shopping assistant. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6465–6472. AAAI Press, Honolulu, USA (2019). https://doi.org/10.1609/aaai.v33i01.33016465

66. González-Caro, C., Baeza-Yates, R.: A multi-faceted approach to query intent classification. In: R. Grossi, F. Sebastiani, F. Silvestri (eds.) String Processing and Information Retrieval, pp. 368–379. Springer, Berlin, Heidelberg (2011)
67. Goo, C.W., Gao, G., Hsu, Y.K., Huo, C.L., Chen, T.C., Hsu, K.W., Chen, Y.N.: Slot-gated modeling for joint slot filling and intent prediction. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pp. 753–757. Association for Computational Linguistics, New Orleans, USA (2018)
68. Griol, D., Callejas, Z.: Statistical dialog tracking and management for task-oriented conversational systems. In: International Conference on Hybrid Artificial Intelligence Systems, pp. 507–518. Springer (2023)
69. Griol, D., Callejas, Z.: Combining statistical dialog management and intent recognition for enhanced response selection. Logic J. IGPL, jzae045 (2024)
70. Gunasekara, C.: Overview of the ninth dialog system technology challenge: Dstc9. In: DSTC9 Workshop at AAAI 2021 (2021)
71. Guo, D., Tür, G., Yih, W.t., Zweig, G.: Joint semantic utterance classification and slot filling with recursive neural networks. In: 2014 IEEE Spoken Language Technology Workshop (SLT), pp. 554–559. IEEE, South Lake Tahoe, USA (2014)
72. Gupta, A., Hewitt, J., Kirchhoff, K.: Simple, fast, accurate intent classification and slot labeling for goal-oriented dialogue systems. In: Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue, pp. 46–55. Association for Computational Linguistics, Stockholm, Sweden (2019). https://doi.org/10.18653/v1/W19-5906. https://www.aclweb.org/anthology/W19-5906
73. Haffner, P., Tur, G., Wright, J.: Optimizing svms for complex call classification. In: 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03), vol. 1, pp. I–I (2003). https://doi.org/10.1109/ICASSP.2003.1198860
74. Hakkani-Tür, D., Tür, G., Celikyilmaz, A., Chen, Y.N., Gao, J., Deng, L., Wang, Y.Y.: Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM. In: Interspeech, pp. 715–719. ISCA, San Francisco, USA (2016)
75. Han, S.C., Long, S., Li, H., Weld, H., Poon, J.: Bi-directional joint neural networks for intent classification and slot filling. In: Proc. Interspeech 2021, pp. 4743–4747. ISCA, Brno, Czech Republic (2021)
76. Haque, A.A.U., Wang, H.: Rethinking conversational recommendations: Is decision tree all you need? In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, pp. 686–695 (2022)
77. Harilal, N., Shah, R., Sharma, S., Bhutani, V.: Caro: an empathetic health conversational chatbot for people with major depression. In: Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, pp. 349–350. Association for Computing Machinery (2020)
78. Hasanuzzaman, M., Saha, S., Dias, G., Ferrari, S.: Understanding temporal query intent. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 823–826 (2015)
79. Hashemi, H.B., Asiaee, A., Kraft, R.: Query intent detection using convolutional neural networks. In: International Conference on Web Search and Data Mining, Workshop on Query Understanding, vol. 23 (2016)
80. Hemphill, C.T., Godfrey, J.J., Doddington, G.R.: The atis spoken language systems pilot corpus. In: Proceedings of the Workshop on Speech and Natural Language, HLT '90, p. 96–101. Association for Computational Linguistics, Hidden Valley, Pennsylvania, USA (1990). https://doi.org/10.3115/116580.116613. https://doi.org/10.3115/116580.116613
81. Henderson, M., Thomson, B., Williams, J.D.: The second dialog state tracking challenge. In: Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), pp. 263–272 (2014)
82. Henderson, M., Thomson, B., Williams, J.D.: The third dialog state tracking challenge. In: 2014 IEEE Spoken Language Technology Workshop (SLT), pp. 324–329. IEEE (2014)

83. Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., Steinhardt, J.: Measuring massive multitask language understanding. In: International Conference on Learning Representations (2020)
84. Hirschman, L.: Multi-site data collection for a spoken language corpus - madcow. In: Second International Conference on Spoken Language Processing (ICSLP'92), pp. 903–906. International Speech Communication Association, Banff, Canada (1992)
85. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
86. Hori, C., Perez, J., Higashinaka, R., Hori, T., Boureau, Y.L., Inaba, M., Tsunomori, Y., Takahashi, T., Yoshino, K., Kim, S.: Overview of the sixth dialog system technology challenge: Dstc6. Comput. Speech Lang. **55**, 1–25 (2019)
87. Huang, Z., Liu, F., Zhou, P., Zou, Y.: Sentiment injected iteratively co-interactive network for spoken language understanding. In: 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7488–7492. IEEE, Toronto, Canada (2021). https://doi.org/10.1109/ICASSP39728.2021.9413885
88. Hui, Y., Wang, J., Cheng, N., Yu, F., Wu, T., Xiao, J.: Joint intent detection and slot filling based on continual learning model. In: 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7643–7647. IEEE, Toronto, Canada (2021). https://doi.org/10.1109/ICASSP39728.2021.9413360
89. IBM: Chatbot for Healthcare - IBM watsonx Assistant — ibm.com. https://www.ibm.com/products/watsonx-assistant/healthcare. [Accessed 24-06-2024]
90. Jacovi, A., El, O.B., Lavi, O., Boaz, D., Amid, D., Ronen, I., Anaby-Tavor, A.: Improving task-oriented dialogue systems in production with conversation logs. In: Converse@ KDD (2020)
91. Jeong, M., Lee, G.G.: Triangular-chain conditional random fields. IEEE Trans. Audio Speech Lang. Process. **16**(7), 1287–1302 (2008)
92. Jin, D., Pan, E., Oufattole, N., Weng, W.H., Fang, H., Szolovits, P.: What disease does this patient have? a large-scale open domain question answering dataset from medical exams. Appl. Sci. **11**(14), 6421 (2021)
93. Jin, Q., Dhingra, B., Liu, Z., Cohen, W., Lu, X.: Pubmedqa: A dataset for biomedical research question answering. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 2567–2577 (2019)
94. Jin, X., Lei, W., Ren, Z., Chen, H., Liang, S., Zhao, Y., Yin, D.: Explicit state tracking with semi-supervisionfor neural dialogue generation. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 1403–1412 (2018)
95. Joshi, R., Chatterjee, A., Ekbal, A.: Towards explainable dialogue system: Explaining intent classification using saliency techniques. In: Proceedings of the 18th International Conference on Natural Language Processing (ICON), pp. 120–127 (2021)
96. Kanhabua, N., Ngoc Nguyen, T., Nejdl, W.: Learning to detect event-related queries for web search. In: Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion, p. 1339–1344. Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2740908.2741698
97. Karat, C.M., Vergo, J., Nahamoo, D.: Conversational interface technologies. The Human-Computer Interaction Handbook, pp. 169–186 (2002)
98. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp. 4171–4186 (2019)
99. Khan, S., Rabbani, M.R.: Chatbot as islamic finance expert (caife) when finance meets artificial intelligence. In: Proceedings of the 2020 4th International Symposium on Computer Science and Intelligent Control, pp. 1–5 (2020)
100. Kim, J.K., Kim, Y.B.: Joint learning of domain classification and out-of-domain detection with dynamic class weighting for satisficing false acceptance rates. In: Proc. Interspeech 2018, pp. 556–560. ISCA, Hyderabad, India (2018). https://doi.org/10.21437/Interspeech.2018-1581

101. Kim, K., Jha, R., Williams, K., Marin, A., Zitouni, I.: Slot tagging for task oriented spoken language understanding in human-to-human conversation scenarios. In: Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), pp. 757–767. Association for Computational Linguistics, Hong Kong, China (2019). https://doi.org/10.18653/v1/K19-1071. https://www.aclweb.org/anthology/K19-1071
102. Kim, S., D'Haro, L.F., Banchs, R.E., Williams, J.D., Henderson, M.: The fourth dialog state tracking challenge. Dialogues with Social Robots: Enablements, Analyses, and Evaluation, pp. 435–449 (2017)
103. Kim, S., D'Haro, L.F., Banchs, R.E., Williams, J.D., Henderson, M., Yoshino, K.: The fifth dialog state tracking challenge. In: 2016 IEEE Spoken Language Technology Workshop (SLT), pp. 511–517. IEEE (2016)
104. Kim, S., Galley, M., Gunasekara, C., Lee, S., Atkinson, A., Peng, B., Schulz, H., Gao, J., Li, J., Adada, M., et al.: Overview of the eighth dialog system technology challenge: Dstc8. IEEE/ACM Trans. Audio Speech Lang. Process. **29**, 2529–2540 (2021)
105. Kim, Y.B., Lee, S., Stratos, K.: ONENET: Joint domain, intent, slot prediction for spoken language understanding. In: 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 547–553. IEEE, Okinawa, Japan (2017)
106. Kirschner, M., Bernardi, R., Baroni, M., Dinh, L.T.: Analyzing interactive qa dialogues using logistic regression models. In: AI* IA 2009: Emergent Perspectives in Artificial Intelligence: XIth International Conference of the Italian Association for Artificial Intelligence Reggio Emilia, Italy, December 9–12, 2009 Proceedings 11, pp. 334–344. Springer (2009)
107. Kong, Y., Zhang, L., Ma, C., Cao, C.: Hsan: A hierarchical self-attention network for multi-turn dialogue generation. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7433–7437. IEEE (2021)
108. Korpusik, M., Liu, Z., Glass, J.: A comparison of deep learning methods for language understanding. In: Proc. Interspeech 2019, pp. 849–853. ISCA, Graz, Austria (2019). https://doi.org/10.21437/Interspeech.2019-1262
109. Krone, J., Zhang, Y., Diab, M.: Learning to classify intents and slot labels given a handful of examples. In: Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, pp. 96–108. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.nlp4convai-1.12. https://aclanthology.org/2020.nlp4convai-1.12
110. Kumar, A., Dikshit, S., Albuquerque, V.H.C.: Explainable artificial intelligence for sarcasm detection in dialogues. Wirel. Commun. Mobile Comput. **2021**(1), 2939334 (2021)
111. Kumar, R., Patidar, M., Varshney, V., Vig, L., Shroff, G.: Intent detection and discovery from user logs via deep semi-supervised contrastive clustering. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1836–1853 (2022)
112. Kurata, G., Xiang, B., Zhou, B., Yu, M.: Leveraging sentence-level information with encoder LSTM for semantic slot filling. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2077–2083. Association for Computational Linguistics, Austin, Texas (2016). https://doi.org/10.18653/v1/D16-1223. https://www.aclweb.org/anthology/D16-1223
113. Lafferty, J., McCallum, A., Pereira, F., et al.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Icml, vol. 1, p. 3. Williamstown, MA (2001)
114. Lange Di Cesare, K., Zouaq, A., Gagnon, M., Jean-Louis, L.: A machine learning filter for the slot filling task. Information **9**(6), 133 (2018)
115. Lee, J., Kim, D., Sarikaya, R., Kim, Y.B.: Coupled representation learning for domains, intents and slots in spoken language understanding. In: 2018 IEEE Spoken Language Technology Workshop (SLT), pp. 714–719. IEEE, Athens, Greece (2018)
116. Li, C., Kong, C., Zhao, Y.: A joint multi-task learning framework for spoken language understanding. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6054–6058. IEEE, Calgary, Canada (2018)

117. Li, C., Li, L., Qi, J.: A self-attentive model with gate mechanism for spoken language understanding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3824–3833 (2018)
118. Li, C., Zhao, Y., Yu, D.: Conditional joint model for spoken dialogue system. In: R. Xu, J. Wang, L.J. Zhang (eds.) Cognitive Computing – ICCC 2019, pp. 26–36. Springer, Cham (2019)
119. Li, H., Arora, A., Chen, S., Gupta, A., Gupta, S., Mehdad, Y.: MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 2950–2962. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.eacl-main.257. https://aclanthology.org/2021.eacl-main.257
120. Li, X., Aitken, W., Zhu, X., Thomas, S.W.: Learning better intent representations for financial open intent classification. In: Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP), pp. 68–77 (2022)
121. Li, Y., Su, H., Shen, X., Li, W., Cao, Z., Niu, S.: Dailydialog: A manually labelled multi-turn dialogue dataset. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 986–995 (2017)
122. Li, Z., Wu, J., Miao, J., Yu, X.: Improve the response diversity of multi-turn dialogue system by combining knowledge. IAENG Int. J. Comput. Sci. **49**(3), 701–709 (2022)
123. Liao, L., Ma, Y., He, X., Hong, R., Chua, T.s.: Knowledge-aware multimodal dialogue systems. In: Proceedings of the 26th ACM International Conference on Multimedia, pp. 801–809 (2018)
124. Lin, T.E., Xu, H.: Deep unknown intent detection with margin loss. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5491–5496. Association for Computational Linguistics, Florence, Italy (2019). https://doi.org/10.18653/v1/P19-1548. https://www.aclweb.org/anthology/P19-1548
125. Liu, B., Lane, I.: Recurrent neural network structured output prediction for spoken language understanding. In: Proceedings of NIPS Workshop on Machine Learning for Spoken Language Understanding and Interactions. Association for Computational Linguistics, Montreal, Canada (2015)
126. Liu, B., Lane, I.: Attention-based recurrent neural network models for joint intent detection and slot filling. In: Interspeech 2016, pp. 685–689. ISCA, San Francisco, USA (2016). https://doi.org/10.21437/Interspeech.2016-1352.
127. Liu, B., Mazumder, S.: Lifelong and continual learning dialogue systems: learning during conversation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 15058–15063 (2021)
128. Liu, Y., Meng, F., Zhang, J., Zhou, J., Chen, Y., Xu, J.: CM-net: A novel collaborative memory network for spoken language understanding. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 1051–1060. Association for Computational Linguistics, Hong Kong, China (2019). https://doi.org/10.18653/v1/D19-1097. https://www.aclweb.org/anthology/D19-1097
129. Louvan, S., Magnini, B.: Exploring named entity recognition as an auxiliary task for slot filling in conversational language understanding. In: Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI, pp. 74–80. Association for Computational Linguistics, Brussels, Belgium (2018). https://doi.org/10.18653/v1/W18-5711. https://www.aclweb.org/anthology/W18-5711
130. Louvan, S., Magnini, B.: Leveraging non-conversational tasks for low resource slot filling: Does it help? In: Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue, pp. 85–91. Association for Computational Linguistics, Stockholm, Sweden (2019). https://doi.org/10.18653/v1/W19-5911. https://www.aclweb.org/anthology/W19-5911
131. Lowe, R., Pow, N., Serban, I., Pineau, J.: The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. arXiv preprint arXiv:1506.08909 (2015)

132. Ma, H., Wang, J., Qian, L., Lin, H.: Han-regru: hierarchical attention network with residual gated recurrent unit for emotion recognition in conversation. Neural Comput. Appl. **33**, 2685–2703 (2021)
133. Ma, X., Zhang, Z., Zhao, H.: Enhanced speaker-aware multi-party multi-turn dialogue comprehension. IEEE/ACM Trans. Audio Speech Lang. Process. **31**, 2410–2423 (2023)
134. Madotto, A., Lin, Z., Zhou, Z., Moon, S., Crook, P.A., Liu, B., Yu, Z., Cho, E., Fung, P., Wang, Z.: Continual learning in task-oriented dialogue systems. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 7452–7467 (2021)
135. Maeng, W., Lee, J.: Designing a chatbot for survivors of sexual violence: Exploratory study for hybrid approach combining rule-based chatbot and ml-based chatbot. In: Proceedings of the Asian CHI Symposium 2021, pp. 160–166 (2021)
136. Manuvinakurike, R., Bui, T., Chang, W., Georgila, K.: Conversational image editing: Incremental intent identification in a new dialogue task. In: Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, pp. 284–295 (2018)
137. Masumura, R., Shinohara, Y., Higashinaka, R., Aono, Y.: Adversarial training for multi-task and multi-lingual joint modeling of utterance intent classification. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 633–639 (2018)
138. McCarty, L.T.: Reflections on "taxman": An experiment in artificial intelligence and legal reasoning. Harv. Law Rev., 837–893 (1977)
139. McTear, M.: Introducing dialogue systems. In: Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots, pp. 11–42. Springer (2021)
140. McTear, M.: Rule-based dialogue systems: Architecture, methods, and tools. In: Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots, pp. 43–70. Springer (2021)
141. Melero Nogués, M.T., et al.: Combining Machine Learning and Rule-Based Approaches in Spanish Syntactic Generation. Universitat Pompeu Fabra (2006)
142. Mesnil, G., Dauphin, Y., Yao, K., Bengio, Y., Deng, L., Hakkani-Tur, D., He, X., Heck, L., Tur, G., Yu, D., Zweig, G.: Using recurrent neural networks for slot filling in spoken language understanding. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(3), 530–539 (2015). https://doi.org/10.1109/TASLP.2014.2383614
143. Mesnil, G., He, X., Deng, L., Bengio, Y.: Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In: Interspeech, pp. 3771–3775. ISCA, Lyon, France (2013)
144. Miao, J., Wu, J.: Multi-turn dialogue model based on the improved hierarchical recurrent attention network. Int. J. Eng. Modell. **34**(2 Regular Issue), 17–29 (2021)
145. Mirza, P., Sudhi, V., Sahoo, S.R., Bhat, S.R.: Illuminer: Instruction-tuned large language models as few-shot intent classifier and slot filler. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pp. 8639–8651 (2024)
146. Mohasseb, A., Bader-El-Den, M., Cocea, M.: Classification of factoid questions intent using grammatical features. ICT Express **4**(4), 239–242 (2018)
147. Mohasseb, A., Bader-El-Den, M., Cocea, M.: Classification of factoid questions intent using grammatical features. ICT Express **4**(4), 239–242 (2018)
148. Morgan, J.N., Sonquist, J.A.: Problems in the analysis of survey data, and a proposal. J. Am. Stat. Assoc. **58**(302), 415–434 (1963)
149. Namazifar, M., Papangelis, A., Tur, G., Hakkani-Tür, D.: Language model is all you need: Natural language understanding as question answering. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7803–7807. IEEE (2021)
150. Ng, M., Coopamootoo, K.P., Toreini, E., Aitken, M., Elliot, K., van Moorsel, A.: Simulating the effects of social presence on trust, privacy concerns & usage intentions in automated bots for finance. In: 2020 IEEE European symposium on security and privacy workshops (EuroS&PW), pp. 190–199. IEEE (2020)

151. Ni, P., Li, Y., Li, G., Chang, V.: Natural language understanding approaches based on joint task of intent detection and slot filling for iot voice interaction. Neural Comput. Appl. **32**, 16149–16166 (2020)
152. Ning, B., Zhao, D., Liu, X., Li, G.: Eags: An extracting auxiliary knowledge graph model in multi-turn dialogue generation. World Wide Web **26**(4), 1545–1566 (2023)
153. Niu, J., Penn, G.: Rationally reappraising ATIS-based dialogue systems. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5503–5507. Association for Computational Linguistics, Florence, Italy (2019)
154. Niu, Z., Zhong, G., Yu, H.: A review on the attention mechanism of deep learning. Neurocomputing **452**, 48–62 (2021). https://doi.org/10.1016/j.neucom.2021.03.091. https://www.sciencedirect.com/science/article/pii/S092523122100477X
155. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Adv. Neural Inf. Process. Syst. **35**, 27730–27744 (2022)
156. Pal, A., Umapathi, L.K., Sankarasubbu, M.: Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In: Conference on Health, Inference, and Learning, pp. 248–260. PMLR (2022)
157. Pallett, D.S., Dahlgren, N.L., Fiscus, J.G., Fisher, W.M., Garofolo, J.S., Tjaden, B.C.: Darpa february 1992 atis benchmark test results. In: Proceedings of the Workshop on Speech and Natural Language, HLT '91, p. 15–27. Association for Computational Linguistics, USA (1992)
158. Pan, W., Chen, Q., Xu, X., Che, W., Qin, L.: A preliminary evaluation of chatgpt for zero-shot dialogue understanding. arXiv preprint arXiv:2304.04256 (2023)
159. Park, Y.: Automatic call quality monitoring using cost-sensitive classification. In: INTERSPEECH, pp. 3085–3088 (2011)
160. Peng, B., Yao, K.: Recurrent neural networks with external memory for language understanding (2015). arXiv:1506.00195
161. Pentyala, S., Liu, M., Dreyer, M.: Multi-task networks with universe, group, and task feature learning. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 820–830. Association for Computational Linguistics, Florence, Italy (2019). https://doi.org/10.18653/v1/P19-1079. https://www.aclweb.org/anthology/P19-1079
162. Purohit, H., Dong, G., Shalin, V., Thirunarayan, K., Sheth, A.: Intent classification of short-text on social media. In: 2015 IEEE International Conference on Smart City/Socialcom/Sustaincom (Smartcity), pp. 222–228. IEEE (2015)
163. Qi, P., Du, N., Manning, C.D., Huang, J.: Pragmaticqa: A dataset for pragmatic question answering in conversations. In: Findings of the Association for Computational Linguistics: ACL 2023, pp. 6175–6191 (2023)
164. Qin, L., Che, W., Li, Y., Wen, H., Liu, T.: A stack-propagation framework with token-level intent detection for spoken language understanding. In: K. Inui, J. Jiang, V. Ng, X. Wan (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019, pp. 2078–2087. Association for Computational Linguistics, Hong Kong (2019). https://doi.org/10.18653/v1/D19-1214
165. Qin, L., Liu, T., Che, W., Kang, B., Zhao, S., Liu, T.: A co-interactive transformer for joint slot filling and intent detection. In: 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8193–8197. IEEE, Toronto, Canada (2021). https://doi.org/10.1109/ICASSP39728.2021.9414110
166. Qin, L., Xu, X., Che, W., Liu, T.: Agif: An adaptive graph-interactive framework for joint multiple intent detection and slot filling. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 1807–1816 (2020)
167. Qiu, L., Chen, Y., Jia, H., Zhang, Z.: Query intent recognition based on multi-class features. IEEE Access **6**, 52195–52204 (2018)
168. Qu, C., Yang, L., Croft, W.B., Zhang, Y., Trippas, J.R., Qiu, M.: User intent prediction in information-seeking conversations. In: Proceedings of the 2019 Conference on Human Information Interaction and Retrieval, pp. 25–33 (2019)

169. Qu, X., Liu, H., Sun, Z., Yin, X., Ong, Y.S., Lu, L., Ma, Z.: Towards building voice-based conversational recommender systems: Datasets, potential solutions and prospects. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2701–2711 (2023)
170. Quan, J., Xiong, D.: Modeling long context for task-oriented dialogue state generation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7119–7124 (2020)
171. Quan, J., Yang, M., Gan, Q., Xiong, D., Liu, Y., Dong, Y., Ouyang, F., Tian, J., Deng, R., Li, Y., et al.: Integrating pre-trained model into rule-based dialogue management. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 16097–16099 (2021)
172. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989). https://doi.org/10.1109/5.18626
173. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
174. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training. OpenAI (2018)
175. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog (2019)
176. Ramshaw, L.A., Marcus, M.P.: Text chunking using transformation-based learning. In: Natural Language Processing Using Very Large Corpora, pp. 157–176. Springer (1999)
177. Rao, M., Raju, A., Dheram, P., Bui, B., Rastrow, A.: Speech to semantics: Improve asr and nlu jointly via all-neural interfaces. In: Interspeech 2020 (2020). https://www.amazon.science/publications/speech-to-semantics-improve-asr-and-nlu-jointly-via-all-neural-interfaces
178. Ravuri, S., Stolcke, A.: Recurrent neural network and LSTM models for lexical utterance classification. In: Interspeech, pp. 135–139. ISCA, Dresden, Germany (2015)
179. Raymond, C., Riccardi, G.: Generative and discriminative algorithms for spoken language understanding. In: Interspeech, pp. 1605–1608. ISCA, Antwerp, Belgium (2007)
180. Ren, F., Xue, S.: Intention detection based on siamese neural network with triplet loss. IEEE Access **8**, 82242–82254 (2020)
181. Rieser, V., Lemon, O.: Using logistic regression to initialise reinforcement-learning-based dialogue systems. In: 2006 IEEE Spoken Language Technology Workshop, pp. 190–193. IEEE (2006)
182. Salehinejad, H., Sankar, S., Barfett, J., Colak, E., Valaee, S.: Recent advances in recurrent neural networks. arXiv preprint arXiv:1801.01078 (2017)
183. Sang, E.T.K., Veenstra, J.: Representing text chunks. In: Ninth Conference of the European Chapter of the Association for Computational Linguistics, pp. 173–179. Association for Computational Linguistics, Bergen, Norway (1999)
184. Sarikaya, R., Crook, P.A., Marin, A., Jeong, M., Robichaud, J.P., Celikyilmaz, A., Kim, Y.B., Rochette, A., Khan, O.Z., Liu, X., et al.: An overview of end-to-end language understanding and dialog management for personal digital assistants. In: 2016 IEEE Spoken Language Technology Workshop (SLT), pp. 391–397. IEEE (2016)
185. Sarikaya, R., Hinton, G.E., Ramabhadran, B.: Deep belief nets for natural language call-routing. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5680–5683. IEEE (2011)
186. Schuster, S., Gupta, S., Shah, R., Lewis, M.: Cross-lingual transfer learning for multilingual task oriented dialog. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 3795–3805. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1380. https://www.aclweb.org/anthology/N19-1380
187. Setyawan, M.Y.H., Awangga, R.M., Efendi, S.R.: Comparison of multinomial naive bayes algorithm and logistic regression for intent classification in chatbot. In: 2018 International Conference on Applied Engineering (ICAE), pp. 1–5. IEEE (2018)

188. Shah, P., Hakkani-Tür, D., Tür, G., Rastogi, A., Bapna, A., Nayak, N., Heck, L.: Building a conversational agent overnight with dialogue self-play. arXiv preprint arXiv:1801.04871 (2018)
189. Shen, Y., Zeng, X., Jin, H.: A progressive model to enable continual learning for semantic slot filling. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 1279–1284. Association for Computational Linguistics, Hong Kong, China (2019). https://doi.org/10.18653/v1/D19-1126. https://www.aclweb.org/anthology/D19-1126
190. Shi, Y., Yao, K., Chen, H., Pan, Y.C., Hwang, M.Y., Peng, B.: Contextual spoken language understanding using recurrent neural networks. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5271–5275. IEEE, Brisbane, Australia (2015)
191. Shin, Y., Yoo, K., Lee, S.g.: Slot filling with delexicalized sentence generation. In: Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2–6 September 2018, pp. 2082–2086. ISCA, Hyderabad, India (2018). https://doi.org/10.21437/Interspeech.2018-1808
192. Shridhar, K., Dash, A., Sahu, A., Pihlgren, G.G., Alonso, P., Pondenkandath, V., Kovács, G., Simistira, F., Liwicki, M.: Subword semantic hashing for intent classification on small datasets. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–6. IEEE (2019)
193. Si, S., Ma, W., Gao, H., Wu, Y., Lin, T.E., Dai, Y., Li, H., Yan, R., Huang, F., Li, Y.: Spokenwoz: A large-scale speech-text benchmark for spoken task-oriented dialogue agents. Adv. Neural Inf. Process. Syst. **36**, 1–31 (2024)
194. Siddhant, A., Goyal, A.K., Metallinou, A.: Unsupervised transfer learning for spoken language understanding in intelligent agents. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 4959–4966. AAAI Press, Honolulu, USA (2019)
195. Singhal, K., Azizi, S., Tu, T., Mahdavi, S.S., Wei, J., Chung, H.W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., et al.: Large language models encode clinical knowledge. Nature **620**(7972), 172–180 (2023)
196. Skantze, G.: Towards a general, continuous model of turn-taking in spoken dialogue using lstm recurrent neural networks. In: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, pp. 220–230 (2017)
197. Song, G., Wang, M.L., Wang, Z.J., Ye, X.D.: A motion intent recognition method for lower limbs based on cnn-rf combined model. In: 2019 IEEE 5th International Conference on Mechatronics System and Robots (ICMSR), pp. 49–53. IEEE (2019)
198. Song, G., Wang, Y., Wang, M., Li, Y.: Lower limb movement intent recognition based on grid search random forest algorithm. In: Proceedings of the 3rd International Conference on Robotics, Control and Automation, pp. 225–229 (2018)
199. Song, J., Zhang, K., Zhou, X., Wu, J.: Hka: A hierarchical knowledge attention mechanism for multi-turn dialogue system. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3512–3516. IEEE (2020)
200. Staliūnaitė, I., Iacobacci, I.: Auxiliary capsules for natural language understanding. In: 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8154–8158. IEEE, Barcelona, Spain (2020)
201. Strobelt, H., Gehrmann, S., Pfister, H., Rush, A.M.: Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. IEEE Trans. Vis. Comput. Graph. **24**(1), 667–676 (2017)
202. Sun, R., Rao, L., Zhou, X.: Bidirectional information transfer scheme for joint intent detection and slot filling. In: 2021 17th International Conference on Computational Intelligence and Security (CIS), pp. 333–337. IEEE, Chengdu, China (2021). https://doi.org/10.1109/CIS54983.2021.00076
203. Sun, Y., Hu, Y., Xing, L., Yu, J., Xie, Y.: History-adaption knowledge incorporation mechanism for multi-turn dialogue system. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 8944–8951 (2020)

204. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International Conference on Machine Learning, pp. 3319–3328. PMLR (2017)
205. Tang, H., Ji, D., Zhou, Q.: End-to-end masked graph-based CRF for joint slot filling and intent detection. Neurocomputing **413**, 348–359 (2020). https://doi.org/10.1016/j.neucom.2019.06.113
206. Tao, C., Gao, S., Shang, M., Wu, W., Zhao, D., Yan, R.: Get the point of my utterance! learning towards effective responses with multi-head attention mechanism. In: IJCAI, pp. 4418–4424 (2018)
207. Tao, S., Qin, Y., Chen, Y., Du, C., Sun, H., Meng, W., Xiao, Y., Guo, J., Su, C., Wang, M., Zhang, M., Wang, Y., Yang, H.: Incorporating complete syntactical knowledge for spoken language understanding. In: B. Qin, Z. Jin, H. Wang, J. Pan, Y. Liu, B. An (eds.) Knowledge Graph and Semantic Computing: Knowledge Graph Empowers New Infrastructure Construction, pp. 145–156. Springer Singapore, Singapore (2021)
208. Tavares, D., Azevedo, P., Semedo, D., Sousa, R., Magalhaes, J.: Task conditioned bert for joint intent detection and slot-filling. In: EPIA Conference on Artificial Intelligence, pp. 467–480. Springer (2023)
209. Thi Do, Q.N., Gaspers, J.: Cross-lingual transfer learning for spoken language understanding. In: 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5956–5960. IEEE, Brighton, United Kingdom of Great Britain and Northern Ireland (2019)
210. Thorat, S.A., Jadhav, V.: A review on implementation issues of rule-based chatbot systems. In: Proceedings of the International Conference on Innovative Computing & Communications (ICICC) (2020)
211. Tian, X., Huang, L., Lin, Y., Bao, S., He, H., Yang, Y., Wu, H., Wang, F., Sun, S.: Amendable generation for dialogue state tracking. In: Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI, pp. 80–92 (2021)
212. Tu, N.A., Uyen, H.T.T., Phuong, T.M., Bach, N.X.: Joint Multiple Intent Detection and Slot Filling with Supervised Contrastive Learning and Self-Distillation. In: Proceedings of the 26th ECAI, pp. 333–343 (2023)
213. Tür, G., Hakkani-Tür, D., Heck, L.: What is left to be understood in ATIS? In: 2010 IEEE Spoken Language Technology Workshop, pp. 19–24. IEEE, Berkeley, USA (2010)
214. Tür, G., Hakkani-Tür, D., Heck, L., Parthasarathy, S.: Sentence simplification for spoken language understanding. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5628–5631. IEEE, Prague, Czech Republic (2011)
215. Vaira, L., Bochicchio, M.A., Conte, M., Casaluci, F.M., Melpignano, A.: Mamabot: a system based on ml and nlp for supporting women and families during pregnancy. In: Proceedings of the 22nd International Database Engineering & Applications Symposium, pp. 273–277 (2018)
216. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Adv. Neural Inf. Process. Syst. **30**, 1–11 (2017)
217. Veyseh, A.P.B., Dernoncourt, F., Nguyen, T.H.: Improving slot filling by utilizing contextual information. In: Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, pp. 90–95 (2020)
218. Vu, T.: Sequential convolutional neural networks for slot filling in spoken language understanding. In: Interspeech, pp. 3250–3254. ISCA, San Francisco, USA (2016). https://doi.org/10.21437/Interspeech.2016-395
219. Vu, T., Gupta, P., Adel, H., Schütze, H.: Bi-directional recurrent neural network with ranking loss for spoken language understanding. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6060–6064. IEEE, Shanghai, China (2016). https://doi.org/10.1109/ICASSP.2016.7472841
220. Wahde, M., Virgolin, M.: Conversational agents: Theory and applications. In: Handbook On Computer Learning and Intelligence: Volume 2: Deep Learning, Intelligent Control and Evolutionary Computation, pp. 497–544. World Scientific (2022)

221. Wang, C., Huang, Z., Hu, M.: SASGBC: Improving sequence labeling performance for joint learning of slot filling and intent detection. In: Proceedings of the 6th International Conference on Computing and Data Engineering, ICCDE 2020, p. 29–33. Association for Computing Machinery, New York, USA (2020)
222. Wang, H., Cui, M., Zhou, Z., Wong, K.F.: Topicrefine: Joint topic prediction and dialogue response generation for multi-turn end-to-end dialogue system. In: Proceedings of the 5th International Conference on Natural Language and Speech Processing (ICNLSP 2022), pp. 19–29 (2022)
223. Wang, Y., Huang, J., He, T., Tu, X.: Dialogue intent classification with character-cnn-bgru networks. Multimedia Tools Appl. **79**(7), 4553–4572 (2020)
224. Wang, Y., Patel, A., Jin, H.: A new concept of deep reinforcement learning based augmented general tagging system. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 1683–1693. Association for Computational Linguistics, Santa Fe, New Mexico, USA (2018). https://www.aclweb.org/anthology/C18-1143
225. Wang, Y., Shen, Y., Jin, H.: A bi-model based RNN semantic frame parsing model for intent detection and slot filling. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pp. 309–314. Association for Computational Linguistics, New Orleans, Louisiana (2018). https://doi.org/10.18653/v1/N18-2050. https://aclanthology.org/N18-2050
226. Wang, Y., Tang, L., He, T.: Attention-based cnn-blstm networks for joint intent detection and slot filling. In: M. Sun, T. Liu, X. Wang, Z. Liu, Y. Liu (eds.) Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, pp. 250–261. Springer, Cham (2018)
227. Wang, Y.Y.: Strategies for statistical spoken language understanding with small amount of data - an empirical study. In: Interspeech, pp. 2498–2501. ISCA, Makuhari, Japan (2010)
228. Wang, Y.Y., Deng, L., Acero, A.: Spoken language understanding: An introduction to the statistical framework. IEEE Signal Process. Mag. **22**(5), 16–31 (2005)
229. Weissenborn, D., Kočiskỳ, T., Dyer, C.: Dynamic integration of background knowledge in neural nlu systems. arXiv preprint arXiv:1706.02596 (2017)
230. Weizenbaum, J.: Eliza—a computer program for the study of natural language communication between man and machine. Commun. ACM **9**(1), 36–45 (1966)
231. Weld, H., Hu, S., Long, S., Poon, J., Han, S.: Tri-level joint natural language understanding for multi-turn conversational datasets. In: Proc. INTERSPEECH 2023, pp. 700–704. Dublin, Ireland (2023). https://doi.org/10.21437/Interspeech.2023-2300
232. Weld, H., Hu, S., Long, S., Poon, J., Han, S.: Tri-level Joint Natural Language Understanding for Multi-turn Conversational Datasets. In: Proceedings of the INTERSPEECH 2023, pp. 700–704 (2023)
233. Weld, H., Huang, G., Lee, J., Zhang, T., Wang, K., Guo, X., Long, S., Poon, J., Han, C.: CONDA: a CONtextual Dual-Annotated dataset for in-game toxicity understanding and detection. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 2406–2416. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.findings-acl.213. https://aclanthology.org/2021.findings-acl.213
234. Wen, L., Wang, X., Dong, Z., Chen, H.: Jointly modeling intent identification and slot filling with contextual and hierarchical information. In: X. Huang, J. Jiang, D. Zhao, Y. Feng, Y. Hong (eds.) Natural Language Processing and Chinese Computing, pp. 3–15. Springer, Cham (2018)
235. Wen, T.H., Gašić, M., Mrkšić, N., Rojas-Barahona, L.M., Su, P.H., Ultes, S., Vandyke, D., Young, S.: Conditional generation and snapshot learning in neural dialogue systems. In: J. Su, K. Duh, X. Carreras (eds.) Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2153–2162. Association for Computational Linguistics, Austin, Texas (2016). https://doi.org/10.18653/v1/D16-1233. https://aclanthology.org/D16-1233

236. Wen, T.H., Vandyke, D., Mrkšić, N., Gašić, M., Rojas-Barahona, L.M., Su, P.H., Ultes, S., Young, S.: A network-based end-to-end trainable task-oriented dialogue system. In: M. Lapata, P. Blunsom, A. Koller (eds.) Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pp. 438–449. Association for Computational Linguistics, Valencia, Spain (2017). https://aclanthology.org/E17-1042
237. Williams, J.D., Henderson, M., Raux, A., Thomson, B., Black, A., Ramachandran, D.: The dialog state tracking challenge series. AI Mag. **35**(4), 121–124 (2014)
238. Williams, J.D., Raux, A., Henderson, M.: The dialog state tracking challenge series: A review. Dialogue Discourse **7**(3), 4–33 (2016)
239. Winograd, T.: Procedures as a Representation for Data in a Computer Program for Understanding Natural Language. Massachusetts Institute of Technology (1971)
240. Wu, T.W., Juang, B.: Infusing Context and Knowledge Awareness in Multi-turn Dialog Understanding. In: Findings of the EACL 2023, pp. 254–264 (2023)
241. Xia, C., Zhang, C., Yan, X., Chang, Y., Philip, S.Y.: Zero-shot user intent detection via capsule neural networks. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3090–3099 (2018)
242. Xie, W., Gao, D., Ding, R., Hao, T.: A feature-enriched method for user intent classification by leveraging semantic tag expansion. In: Natural Language Processing and Chinese Computing: 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26–30, 2018, Proceedings, Part II 7, pp. 224–234. Springer (2018)
243. Xing, B., Tsang, I.: Co-guiding net: Achieving mutual guidances between multiple intent detection and slot filling via heterogeneous semantics-label graphs. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 159–169. Abu Dhabi (2022)
244. Xu, C., Li, Q., Zhang, D., Cui, J., Sun, Z., Zhou, H.: A model with length-variable attention for spoken language understanding. Neurocomputing **379**, 197–202 (2020)
245. Xu, P., Sarikaya, R.: Convolutional neural network based triangular CRF for joint intent detection and slot filling. In: Workshop on Automatic Speech Recognition and Understanding, pp. 78–83. IEEE, Vancouver, Canada (2013)
246. Xu, P., Sarikaya, R.: Exploiting shared information for multi-intent natural language sentence classification. In: Interspeech, pp. 3785–3789. ISCA, Lyon, France (2013)
247. Xu, W., Haider, B., Mansour, S.: End-to-end slot alignment and recognition for cross-lingual NLU. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5052–5063. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.emnlp-main.410. https://aclanthology.org/2020.emnlp-main.410
248. Xu, Y., Zhao, H., Zhang, Z.: Topic-aware multi-turn dialogue modeling. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 14176–14184 (2021)
249. Yamaguchi, H., Mozgovoy, M., Danielewicz-Betz, A.: A chatbot based on aiml rules extracted from twitter dialogues. In: FedCSIS (Communication Papers), pp. 37–42 (2018)
250. Yang, X., Chen, Y.N., Hakkani-Tür, D., Crook, P., Li, X., Gao, J., Deng, L.: End-to-end joint learning of natural language understanding and dialogue manager. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5690–5694. IEEE, New Orleans, USA (2017)
251. Yao, K., Peng, B., Zhang, Y., Yu, D., Zweig, G., Shi, Y.: Spoken language understanding using long short-term memory neural networks. In: 2014 IEEE Spoken Language Technology Workshop (SLT), pp. 189–194. IEEE, South Lake Tahoe, USA (2014). https://doi.org/10.1109/SLT.2014.7078572
252. Yao, K., Peng, B., Zweig, G., Yu, D., Li, X., Gao, F.: Recurrent conditional random field for language understanding. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4077–4081. IEEE, Florence, Italy (2014)
253. Yao, K., Zweig, G., Hwang, M.Y., Shi, Y., Yu, D.: Recurrent neural networks for language understanding. In: Interspeech, pp. 2524–2528. ISCA, Lyon, France (2013). https://doi.org/10.13140/2.1.2755.3285

254. Ye, F., Manotumruksa, J., Zhang, Q., Li, S., Yilmaz, E.: Slot self-attentive dialogue state tracking. In: Proceedings of the Web Conference 2021, pp. 1598–1608 (2021)
255. Yilmaz, E.H., Toraman, C.: Kloos: Kl divergence-based out-of-scope intent detection in human-to-machine conversations. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, p. 2105–2108. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3397271.3401318
256. Yoshino, K., Chen, Y.N., Crook, P., Kottur, S., Li, J., Hedayatnia, B., Moon, S., Fei, Z., Li, Z., Zhang, J., et al.: Overview of the tenth dialog system technology challenge: Dstc10. IEEE/ACM Trans. Audio Speech Lang. Process. **32**, 765–778 (2023)
257. Yu, D., Wang, S., Deng, L.: Sequential labeling using deep-structured conditional random fields. IEEE J. Sel. Top. Signal Process. **4**(6), 965–973 (2010). https://doi.org/10.1109/JSTSP.2010.2075990
258. Yu, D., Wang, S., Deng, L.: Sequential labeling using deep-structured conditional random fields. IEEE J. Sel. Top. Signal Process. **4**, 965–973 (2011). https://doi.org/10.1109/JSTSP.2010.2075990
259. Yu, S., Chen, Y., Zaidi, H.: Ava: A financial service chatbot based on deep bidirectional transformers. Front. Appl. Math. Stat. **7**, 604842 (2021)
260. Yu, S., Shen, L., Zhu, P., Chen, J.: ACJIS: A novel attentive cross approach for joint intent detection and slot filling. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–7. IEEE, Rio de Janeiro, Brazil (2018)
261. Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: Lstm cells and network architectures. Neural Comput. **31**(7), 1235–1270 (2019)
262. Yulan He, Young, S.: A data-driven spoken language understanding system. In: 2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No.03EX721), pp. 583–588. IEEE, Piscataway, USA (2003)
263. Zang, X., Rastogi, A., Gupta, R., Chen, J.J., Sunkara, S., Zhang, J.: Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. In: Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, pp. 109–117 (2020)
264. Zhang, C., Fan, W., Du, N., Yu, P.S.: Mining user intentions from medical queries: A neural network based heterogeneous jointly modeling approach. In: Proceedings of the 25th International Conference on World Wide Web, WWW '16, p. 1373–1384. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2016). https://doi.org/10.1145/2872427.2874810
265. Zhang, C., Li, Y., Du, N., Fan, W., Yu, P.: Joint slot filling and intent detection via capsule neural networks. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5259–5267. Association for Computational Linguistics, Florence, Italy (2019). https://doi.org/10.18653/v1/P19-1519. https://www.aclweb.org/anthology/P19-1519
266. Zhang, H., Lan, Y., Pang, L., Guo, J., Cheng, X.: Recosa: Detecting the relevant contexts with self-attention for multi-turn dialogue generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3721–3730 (2019)
267. Zhang, L., Ma, D., Zhang, X., Yan, X., Wang, H.F.: Graph lstm with context-gated mechanism for spoken language understanding. In: AAAI 2020, pp. 9539–9546. AAAI Press, New York, USA (2020)
268. Zhang, L., Wang, H.: Using bidirectional transformer-crf for spoken language understanding. In: J. Tang, M.Y. Kan, D. Zhao, S. Li, H. Zan (eds.) Natural Language Processing and Chinese Computing, pp. 130–141. Springer, Cham (2019)
269. Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., Weston, J.: Personalizing dialogue agents: I have a dog, do you have pets too? In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2204–2213 (2018)

270. Zhang, S., Jiang, J., He, Z., Zhao, X., Fang, J.: A novel slot-gated model combined with a key verb context feature for task request understanding by service robots. IEEE Access **7**, 105937–105947 (2019)
271. Zhang, W., Li, Z., Guo, Y.: Leveraging different context for response generation through topic-guided multi-head attention. In: Proceedings of the 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence, pp. 1–6 (2020)
272. Zhang, X., Wang, H.: A joint model of intent determination and slot filling for spoken language understanding. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), pp. 2993–2999. AAAI Press, New York, USA (2016)
273. Zhang, Z., Huang, H., Wang, K.: Using deep time delay neural network for slot filling in spoken language understanding. Symmetry **12**(6), 1–17 (2020)
274. Zhang, Z., Li, J., Zhao, H.: Multi-turn dialogue reading comprehension with pivot turns and knowledge. IEEE/ACM Trans. Audio Speech Lang. Process. **29**, 1161–1173 (2021)
275. Zhang, Z., Zhang, Z., Chen, H., Zhang, Z.: A joint learning framework with BERT for spoken language understanding. IEEE Access **7**, 168849–168858 (2019)
276. Zhao, L., Feng, Z.: Improving slot filling in spoken language understanding with joint pointer and attention. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 426–431. Association for Computational Linguistics, Melbourne, Australia (2018). https://doi.org/10.18653/v1/P18-2068. https://www.aclweb.org/anthology/P18-2068
277. Zheng, Y., Liu, Y., Hansen, J.H.L.: Intent detection and semantic parsing for navigation dialogue language processing. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 1–6. IEEE, Yokohama, Japan (2017)
278. Zhou, P., Huang, Z., Liu, F., Zou, Y.: Pin: A novel parallel interactive network for spoken language understanding. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 2950–2957. IEEE, Los Alamitos, USA (2021). https://doi.org/10.1109/ICPR48806.2021.9411948. https://doi.ieeecomputersociety.org/10.1109/ICPR48806.2021.9411948
279. Zhou, Q., Wen, L., Wang, X., Ma, L., Wang, Y.: A hierarchical lstm model for joint tasks. In: M. Sun, X. Huang, H. Lin, Z. Liu, Y. Liu (eds.) Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, pp. 324–335. Springer, Cham (2016)
280. Zhu, S., Yu, K.: Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5675–5679. IEEE, New Orleans, USA (2017). https://doi.org/10.1109/ICASSP.2017.7953243
281. Zhu, S., Zhao, Z., Ma, R., Yu, K.: Prior knowledge driven label embedding for slot filling in natural language understanding. IEEE/ACM Trans. Audio Speech Lang. Process. **28**, 1440–1451 (2020). https://doi.org/10.1109/TASLP.2020.2980152
282. Zhu, Z., Cheng, X., An, H., Wang, Z., Chen, D., Huang, Z.: Zero-shot spoken language understanding via large language models: A preliminary study. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pp. 17877–17883 (2024)
283. Zhu, Z., Xu, W., Cheng, X., Song, T., Zou, Y.: A dynamic graph interactive framework with label-semantic injection for spoken language understanding. In: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5. Rhodes Island, Greece (2023). https://doi.org/10.1109/ICASSP49357.2023.10095809
284. Zhu, Z., Xu, W., Cheng, X., Song, T., Zou, Y.: A dynamic graph interactive framework with label-semantic injection for spoken language understanding. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5. IEEE, Rhodes Island, Greece (2023)