Università di Catania

Dipartimento di Scienze Umanistiche
Dottorato di Ricerca in Scienze dell'Interpretazione

SABRINA MENNELLA

Speaking less to say more:
A Linguistic Approach Investigating Commonsense Knowledge for the Implementation of Conversational Agents

TUTOR
Prof. Marco Venuti

CO-TUTOR
Prof. Francesco Cutugno

XXXVII CICLO

Speaking less to say more: A Linguistic Approach Investigating Commonsense Knowledge for the Implementation of Conversational Agents

Sabrina Mennella

A dissertation presented for the degree of
Doctor of Philosophy

Tutor: Prof. Marco Venuti
Co-Tutor: Prof. Francesco Cutugno

Department of Humanities
University of Catania
Italy

2025

"*Less is more.*"

Ludwig Mies van der Rohe

Speaking less to say more:

A Linguistic Approach Investigating Commonsense Knowledge for the Implementation of Conversational Agents

by Sabrina MENNELLA

ABSTRACT

This work focuses on Commonsense Knowledge from a linguistic perspective to develop a theoretical framework supporting the implementation of conversational agents. Commonsense Knowledge is described as a complex and multifaceted structure, encompassing a wide range of knowledge generally acquired through everyday experiences. It is often implicit in communication (written or oral), posing a challenge for Natural Language Processing systems that leverage a data-driven approach to simulate human knowledge acquisition information. Therefore, its multifaceted nature suggests its conceptualisation as a *process* rather than a static information collection. Through a three-level analysis, the main goal is to investigate information structure starting from linguistic data. Together with computational tools, such as graph databases, an efficient data analysis has been made possible, revealing recurrent linguistic patterns. The advantage of this model in a human-computer interaction application lies in allowing the system to automatically omit redundant information during the communicative interaction.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	**A**rtificial **I**ntelligence
ASR	**A**utomatic **S**peech **R**ecogniser
BN	**B**ayesian **N**etworks
BT	**B**ehaviour **T**rees
CK	**C**ommon **K**nowledge
CG	**C**ommon **G**round
CCG	**C**ommunal **C**ommon **G**round
CS	**C**ommon **S**ense
CSK	**C**ommon **S**ense **K**nowledge
CSR	**C**ommonsense **R**easoning
CRUD	**C**reate, **R**ead, **U**pdate, and **D**elete
DM	**D**ialogue **M**anager
DP	**D**ialogue **P**olicy
DST	**D**ialogue **S**tate **T**racking
EK	**E**ncyclopedic **K**nowledge
FE	**F**rame **E**lement
FI	**F**rame **I**ntent
GDBMS	**G**raph **D**atabase **M**anagement **S**ystem
KB	**K**nowledge **B**ases
KG	**K**nowledge **G**raph
KG	**K**nowledge **R**epresentation
LLMs	**L**arge **L**anguage **M**odels
LCG	**L**ocal **C**ommon **G**round
LSTM	**L**ong **S**hort-**T**erm **M**emory
NER	**N**amed **E**ntity **R**ecognition

NLG	**N**atural **L**anguage **G**eneration
NLI	**N**atural **L**anguage **I**nference
NLP	**N**atural **L**anguage **P**rocessing
NLU	**N**atural **L**anguage **U**nderstanding
NLM	**N**eural **L**anguage **M**odel
ODD	**O**pen-domain **D**ialogue **S**ystem
POS	**P**art-**O**f-**S**peech
PCG	**P**ersonal **C**ommon **G**round
PTLM	**P**re-**T**rained **L**anguage **M**odel
Q&A	**Q**uestion **A**nswering
RDMS	**R**elational **D**atabase **M**anagement **S**ystem
RTE	**R**ecognizing **T**extual **E**ntailment
RNNs	**R**ecurrent **N**eural **N**etworks
SCG	**S**pecialised **C**ommon **G**round
SQL	**S**tructured **Q**uery **L**anguage
TOD	**T**ask-**O**riented **D**ialogue **S**ystem
ToM	**T**heory **O**f **M**ind
TMS	**T**ruth **M**aintenance **S**ystem
WSC	**W**inograd **S**chema **C**hallenge

INTRODUCTION

The development of conversational agents aims to create systems capable of engaging in fluid, human-like dialogue. To truly mimic human conversation, these systems require more than just linguistic processing capabilities; they must be able to understand and reason with the vast body of everyday knowledge that humans naturally rely on, known as *Common Sense*. In this respect, equipping machines with Common Sense (CS) is recognised as a central challenge in the Artificial Intelligence (AI) and Natural Language Processing (NLP) field [68, 130], as the extraction and learning algorithms cannot rely on Commonsense Knowledge (CSK), i.e. a set of knowledge, belief and everyday facts widely shared and accessible to speakers that are not explicitly stated in written or oral communication as are taken for granted [125, 48, 146]. In this context, several problems arise mainly concerning the definition of what is meant by CS and how CSK can be efficiently represented to be used by conversational systems to implement their communicative performance [336]. Knowledge Representation (KR) plays a key role in this field, as it deals with how knowledge can be interpreted and manipulated by computers [40]. Ontologies, cornerstones of the Semantic Web [33], have provided for a long time a foundation for structured CSK as essential for representing real-world knowledge. Notably, researchers have constructed large-scale CSK databases, such as CYC [181], ConceptNet [190], ATOMIC [271] and others. The emergence of Generative AI in late 2022 has ignited a paradigm shift, accomplishing specific tasks without reliance on external knowledge or experiences [274]. Unlike previous models that were limited to solving specific tasks, Large Language Models (LLMs) can solve diverse tasks as they possess inferential capacity and extensive reservoirs of knowledge, allowing them to dynamically adjust to the conversational context [9, 53]. However, while LLMs have shown excellent performance in many contextual understanding tasks, including Question Answering (Q&A) and language

generation, they still struggle with some types of Commonsense Reasoning (CSR) [294], producing errors in mathematics [39] and hallucinations [18]. The representation and organisation of CSK, despite significant efforts, remain limited in capturing its full richness and complexity, as the vastness of this knowledge is extensive and impossible to integrate into a repository. The ontology's inherent rigidity still presents a significant challenge as they struggle to adapt to dynamic and evolving domains, hindering their ability to represent complex data structures effectively [171, 45]. As early as the beginning of the '90s, this representation was disputed by Rodney Brooks [43], a leading figure in the field of AI and robotics, who argued that [it is] *better to use the world as its own model*, thus rejecting the possibility of being able to create a database that could contain information capable of capturing all aspects of the human knowledge. This is particularly true for CS, as a comprehensive list of all facts that are obvious to all humans cannot be exhaustively compiled.

To overcome this problem, the perspective is in this work switched, conceiving this knowledge as a *process* that generates structured relationships between entities resulting from recurrent interaction, rather than a static representation of information. In this regard, Saba [264, 265] pointed out that a CSK structure should be discovered rather than created, highlighting that it is insufficient to merely establish a few principles for ontological design but it is essential to develop a strategy to recover its structure. In this work, CSK is not necessarily represented by relationships established in a knowledge base but can be derived and reconstructed using probabilistic approaches, becoming any kind of graphical representation that a speaker believes exists in the knowledge possessed by the interlocutor with a high probability. A three-level analysis model is proposed, employing a probability estimation applied to data collected in a graph database (Neo4J) [323] and describing their linguistic characteristics through a cognitive linguistic approach, such as Frame Semantics [101, 16].

The thesis is structured as follows:

Chapter One provides an introduction to Common Sense in Artificial Intelligence field and its developments through history. This section will examine the representation of Commonsense Knowledge from a technological perspective over the years, tracing its evolution from the development of Knowledge Bases to Large Language

Models. It will specifically address the necessity of such knowledge for dialogue systems to achieve effective conversational performance. To this end, it will illustrate all the components of natural language that aid human understanding. The chapter will conclude by articulating the motivations driving this work, outlining the three research questions in detail.

Chapter Two explores key concepts in cognitive linguistics, with a particular emphasis on cognitive semantics. The notions of *Semantic Frame*, *Cognitive Frame*, and *Communicative Frame* will be examined, since each operates at the levels of language, thought, and communication, respectively. The chapter then shifts focus to cognitive processes in communication, discussing the Theory of Mind, Grice's maxims, and presuppositions. These topics are essential as they provide the theoretical foundation for the framework presented in Chapter 4.

Chapter Three investigates the concept of *Common Sense*, a term that has been studied and analysed in various fields throughout history, such as philosophy. The development of AI appears to be situated at the intersection of 18th-century Scottish/English Enlightenment philosophy and Vico's proposal, which conceptualizes Common Sense as an umbrella term encompassing *Commonsense Knowledge* and *Commonsense Reasoning*, a *communal sensibility* that unites people through shared attention and intentionality. Without delving into an exhaustive exploration of the philosophical background, it will be outlined the general principles to establish a basis for understanding Common Sense, which shares some aspects with concepts pertaining to linguistics, such as Encyclopedic Knowledge and Common Ground.

Chapter Four presents the Commonsense Knowledge hypothesis as a process and distinguishes knowledge into three types: (i) *Background Knowledge*; (ii) *Foreground Knowledge*; and (iii) *Presupposed Knowledge*. Building on this categorisation, the three-level analysis will be defined, and each of these knowledge types will be explored. To carry out the analysis, the cooking domain is taken into account for (i) its ubiquitous familiarity and (ii) the presence of implicit systematic chains of actions in the cooking process. Four domain-specific datasets along with an introduction to the employed tools for the knowledge representation will be detailed.

Chapter Five presents the analyses conducted according to the three levels of analysis. The first level pertains to the identification of domain frames from the FrameNet lexical database and examining their distribution within a dialogic flow. To facilitate this analysis, CookDial, a dialogic corpus comprising 260 dialogues between humans in the cooking domain, is employed and manually annotated using Label Studio, a tool designed for linguistic annotation. The second level of analysis concerns in retrieving and linking domain resources — specifically, Recipe1M+ (a recipe dataset), FlavorDB (a food flavour dataset) and Epic Kitchens (containing recurrent actions performed in the kitchen) - within Neo4J, a graph database employed in this study for the domain representation. In this respect, the most recurrent action extracted from Epic Kitchens on a specific ingredient category belonging to FlavorDB has been identified, providing a deep investigation of the data. The third level of analysis consists of a probabilistic estimation of the Epic Kitchens dataset. Sequences with a high probability of co-occurrence have been extracted, representing chains of frequently performed actions associated with specific objects, employing Markov chains. This approach facilitated conducting a descriptive statistics analysis of the data to examine the linguistic aspects that distinguish verbalized actions from those left implicit.

Chapter Six presents the FANTASIA architecture, a plugin for Unreal Engine designed to facilitate the creation of embodied conversational agents. The discussion will focus on Bastian as an example application for the process of identifying CS instructions that can be omitted to follow a principle of maximum utility.

Chapter Seven concludes the work by discussing two key contributions of this work, emphasizing the importance of linguistic analysis to facilitate the implementation of communicative interaction in human-machine systems.

PUBLICATIONS

- Mennella, S., Di Maro, M., Di Bratto, M. (2024). Estimating Commonsense Knowledge from a Linguistic Analysis on Information Distribution. In Proceedings of the Sixth International Conference on Computational Linguistics in Bulgaria (CLIB 2024), pp. 257-263.

- Mennella, S. (2024). Semantic annotation for extracting Commonsense Knowledge information structure. Il dialogo tra scienze linguistiche e nuove tecnologie (PHD CLUB, Bologna), p. 41.

- Di Bratto, M., Origlia, A., Di Maro, M., Mennella, S. (2024). Linguistics-based dialogue simulations to evaluate argumentative conversational recommender systems. User Modeling and User-Adapted Interaction, pp. 1-31.

- Mennella, S., Di Maro, M., Di Bratto, M. (2023). Common Sense Knowledge graph generation for information-gap requests in dialogue systems. In Proceedings of The 16th International Cognitive Linguistics Conference HHU Düsseldorf, pp. 480-481.

- Origlia, A., Di Bratto, M., Di Maro, M., Mennella, S. (2022). Developing embodied conversational agents in the unreal engine: the FANTASIA Plugin. In Proceedings of the 30th ACM International Conference on Multimedia, pp. 6950-6951.

- Origlia, A., Di Bratto, M., Di Maro, M., Mennella, S. (2022). A multi-source graph representation of the movie domain for recommendation dialogues analysis. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pp. 1297-1306.

CHAPTER

ONE

STATE OF THE ART

This chapter presents the theoretical framework central to this work. The concept of *Common Sense* is introduced along with its critical role in the field of Artificial Intelligence. Specifically, the focus is on *Commonsense Knowledge* and how it is represented in the available Knowledge Bases, with particular emphasis on its application in dialogue systems to enhance understanding and improve natural interactions. Since this knowledge is often taken for granted and not made explicit in texts, the discussion will encompass various Natural Language Processing approaches for retrieving it, highlighting the challenges and limitations in practical applications. These limitations will serve as the foundation for presenting the research proposal that will be outlined at the end of the chapter.

1.1 A Brief Overview

The development of high-quality Artificial Intelligence (AI) hinges on the ambitious goal of equipping machines with *human-level intelligence*, which pertains to a broad spectrum of capabilities including speaking, learning, reasoning, problem-solving and adaptation [330]. The idea of creating intelligent machines has seen multiple inventions throughout history, culminating in the early 50s with the Turing Test [309]. In this work, Alan Turing, regarded as the father of computer science, introduced a test based on an imitation game, in which a human interrogator attempts to differentiate between the text responses of a computer and a human. The main goal was to demonstrate that building a computerised system able to mimic human intelligence was possible and mislead the human operator by making them believe they were

interacting with a *thinking* user. The concept of the *thinking machine* generated significant interest during that period, both in computational and cognitive fields [274]. In 1956, the computer scientist John McCarthy along with the American cognitive and computer scientist Marvin Minsky, wrote a computer program called *Advice Taker* [200]. This formal logic-based program reproduces the processes of inferences about the type and essence of ordinary situations that human beings encounter every day. Their goal was to build a machine able to acquire knowledge through experience, as happens in humans, and capable of inferring any type of information and drawing logical conclusions. In particular, they state:

> *[a] program has Common Sense if it automatically deduces for itself a sufficiently wide class of immediate consequences of anything it is told and what it already knows.* [p.78]

where CS denotes here the ability to infer conclusions, i.e., to deduce implications from what is already known (memory stock) and from information given (input flow) [25][p.13]. Thus, CS serves as an umbrella term to encompass both *Commonsense Knowledge (CSK)*, i.e stock of knowledge comprising a large number of facts and beliefs about the world which is self-evident, requires no logical or other demonstration and is widely shared and accessible among people [48, 257] and *Commonsense Reasoning (CSR)*, i.e. the process of gathering information about specific elements of a scenario and drawing inferences about other aspects based on CSK or an understanding of how the world works [219][p.1]. These two aspects play a crucial role in human communication, as they serve as the foundation to interpret information in everyday interactions. In this respect, developing a conversational intelligence system with CS capabilities has been a main goal in the Natural Language Processing (NLP) field, a branch of AI that focuses on the interaction between computers and humans through natural language [143, 332]. At the core of this advancement lies the role of *conversational agents*, also known as *dialogue systems*. The main aim is to emulate human-to-human conversations, thus offering the possibility of improving the user experience, simplifying tasks and providing personalised assistance in domains such as customer care, virtual assistants, healthcare, education, and others [320].

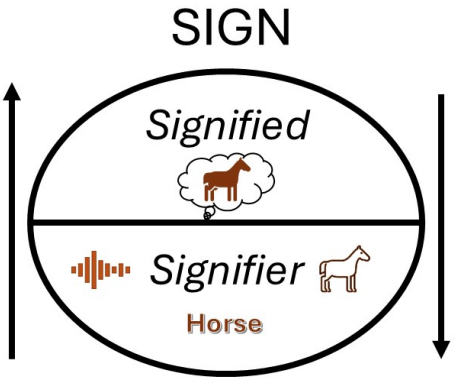

FIGURE 1.1: According to Saussure, the linguistic sign encompasses anything that conveys meaning. The *signifier*, such as words (written/oral) or images, provides the means to convey that meaning (in this example, a horse). In turn, the *signified* represents the mental concept evoked in the mind.

As reported in the Collins Dictionary, the term *communicate* is an act of *sharing or exchanging information with people by speaking, or writing* [1]. Human communication fundamentally relies on *verbal language*, an innate ability of *Homo sapiens* and one of the available tools, modalities, and systems for conveying information which employs *signs* [109][p.5]. According to a widely accepted view among scholars in semiotics and semiology [240, 69], anything can convey meaning. The cultural phenomenon as long as natural events can be perceived and interpreted through human experience, thereby transmitting information [109]. In the Saussurian point of view, a *linguistic sign* is composed of an acoustic or visual *signifier* and a conceptual *signified*; for example, a sign is the acoustic image of the word 'horse', i.e. the signifier of the concept of a horse (the signified) [308] (Figure 1.1). In communication, a sender transmits a sign to a receiver. The receiver's ability to interpret the sign stems from the fact that the sign can be traced back to a shared *code*, i.e. a set of conventions and knowledge that enables meaning to be ascribed to what occurs. This *code* refers more precisely to the set of correspondences, fixed by mutual agreement, between one thing and another, which provides the rules for interpreting signs. All communication systems operate based on such codes, and from this perspective, linguistic signs constitute the *language code*, which includes several properties, including arbitrariness [109][pp.6-7]. In general terms, arbitrariness means that there

[1]Collins Dictionary: https://www.collinsdictionary.com/dictionary [last visited on 26/11/2024]

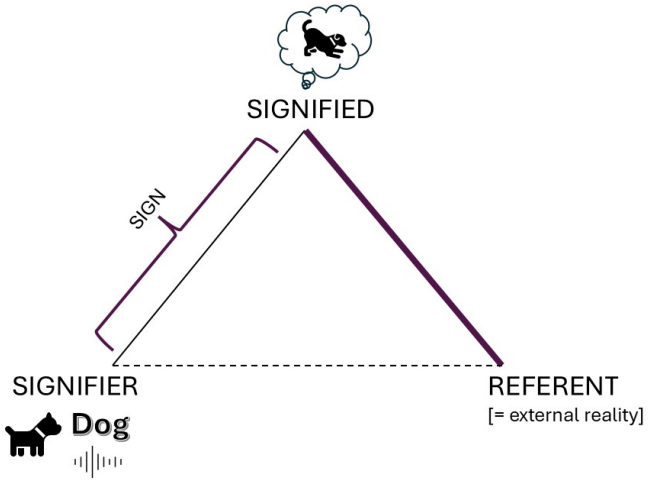

FIGURE 1.2: The Semiotic triangle. The word *dog*, formed by the two sides of the signifier (i.e. d-o-g), and the meaning (i.e."dog"), refers to and identifies the real object **dog**. The triangle's baseline is dotted (as opposed to the two sides) as the relationship between the signifier and the referent is not straight, but is mediated by the signified. (Source: Berruto & Cerruti, 2011 [109])

is no naturally motivated link between the signifier and the meaning of a sign, i.e. there is no direct connection with the nature or essence of things, based on empirical observation or logical reasoning [109][p.8]. For instance, the signifier *horse* has no inherent connection to the animal it represents, as there is nothing intrinsic to the nature of the thing that necessitates its name. However, this does not imply a complete lack of relationships between the signifier and the meaning of the sign. Rather, the links and relationships that do exist - which constitute the underlying code - are established through convention. However, the issue is much more complex than explained so far. In this respect, to address the problem is necessary to take into account the semiotic triangle proposed by Charles Kay Ogden and Ivor Armstrong Richards [225], highlighting the three entities involved in the linguistic signs[2], as shown in Figure 1.2. At the three corners of the triangle, there are three main entities: a *signifier* that, through the mediation of an associated *signified* which it transmits (and together with which it forms the *sign*), refers to an element of external, extralinguistic reality, i.e. a *referent* [109].

[2]It is important to note that there is still some controversy surrounding the interpretation of the semiotic triangle, as not all scholars agree on the identification of the entities represented at its three vertices [109][p.8]. Given the purely informative purpose of this section, I took into account the semiotic triangle definition outlined in Berruto and Cerruti's work [109], which employs Saussurian terminology.

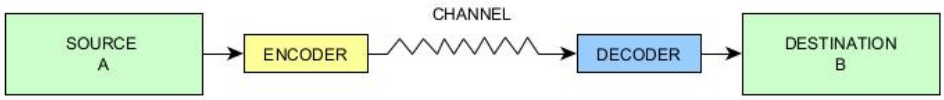

FIGURE 1.3: Shannon-Weaver model of communication.

Earlier communication theories claimed that communication is achieved by encoding and decoding messages based on a single model known as the *code model* [285]. The classical example is represented by the Shannon-Weaver diagram [278] (Figure 1.3). Inspired by telecommunications technology, it displays how communication can take place through the use of a code. A *code* is defined as a system that couples messages and signals, enabling two information processing devices (organisms or machines) to communicate. A *message* is an internal representation of the communicating devices; a *signal* is a change in the external environment that can be produced by one device and recognised by the other. This model shows how a message from an information source can be duplicated to a destination as a result of a communication process. In this context, communication is achieved by encoding a message (which cannot travel) that is transformed into a signal (which can travel) and decoding this signal at the receiving end. If the devices work and the codes are identical at both ends, communication is guaranteed. Otherwise, noise along the channel can destroy or distort the signal, interrupting the communication [308]. The code model is effective in explaining communication in information theory and there is well-known neuroscientific evidence on language processing in the brain that can be interpreted to support this model [308]. The main areas traditionally considered to be involved in language processing are located in the dominant hemisphere, in which can be found Broca's area, located in the left inferior frontal gyrus and Wernicke's area, located in the left posterior superior temporal gyrus [308]. In this sense, it can be argued that human communication consists of transmitting encoded information, assuming that certain information is somehow 'contained' within the mind of subject *A* in the form of mental representations. The linguistic capabilities of *A*, mediated through *A*'s Broca's area, enable *A* to encode a message as a signal emitted by *A*'s speech production system. This signal then propagates through the air as a wavelength until it reaches the auditory system of subject *B*. Lastly, with the linguistic abilities of *B*'s Wernicke area, *B* decodes the signal and generates a mental representation with content analogous to *A*'s original mental state [308][p.15].

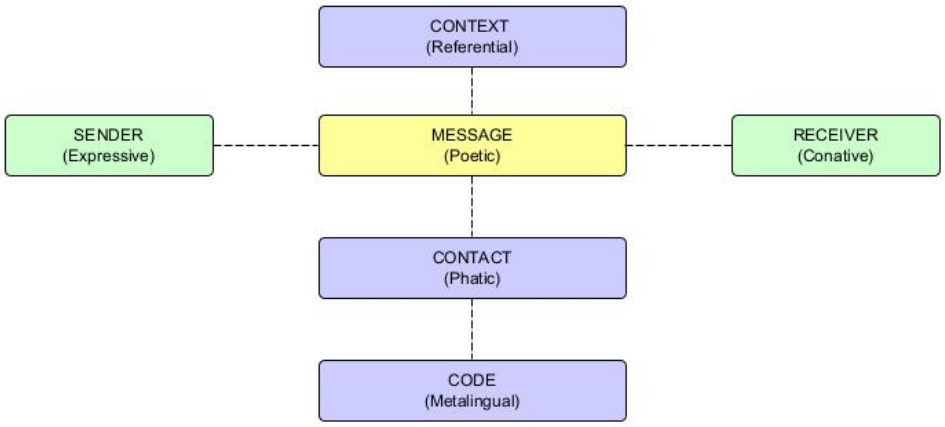

FIGURE 1.4: Jakobson's Model of Language Functions.

In this respect, Jakobson [148] proposed a framework for the functions of language, which aligns with the Shannon-Weaver model, shedding light on the purposes and mechanisms of human communication beyond the simple transmission of data. According to Jakobson, elements involved in communication comprise:

1. the *addresser*, who conveys a message to the receiver;
2. *the message*, which is interpretable and linked to a context to which it refers;
3. the *code* shared by both the addresser and the receiver, which enables the encoding and decoding of the message;
4. the *contact*, which refers to both the physical channel and the psychological connection between addresser and receiver, which facilitates and sustains communication.

Each component of this communicative circuit corresponds to a specific function of language [148, 73], which a schematisation is shown in Figure 1.4:

- **Context** The referential function pertains to the referent described in the message or to which the message refers, such as a situation, object, or mental state. This function is expressed through tools like definite descriptions and deictic terms;
- **Message** The poetic function focuses on the message itself, playing a key role in poetry and slogans, where the form and aesthetic impact of language are emphasized.

- **Addresser** The emotive function reflects the addresser's internal state and is conveyed through interjections and vocalizations that provide additional information about the speaker's emotions.

- **Addressee** The conative function engages the addressee directly, utilizing forms such as vocatives and imperatives to influence or prompt a response.

- **Contact** The phatic function concerns language used to establish, maintain, verify, or close the communication channel, also known as the contact. This function ensures the smooth transmission of information by addressing potential disruptions, such as environmental noise or misunderstandings. This aspect is commonly explored in pragmatics, particularly in processes related to grounding and verifying mutual understanding.

- **Code** The metalingual or metalinguistic function involves using language to discuss or clarify language itself, facilitating understanding of the code shared by communicators.

However, human communication is not just a matter of exchanging information as the use of codes is not sufficient to characterize what human beings do when they engage in communicative exchanges [308]. In Jean Allwood's work [3], criticism emerges towards the Shannon-Weaver communication model, arguing that it oversimplifies the communication process by focusing too heavily on the technical aspects of sending and receiving messages. Specifically, the author defines human communication as the "*sharing of information involving at least two human beings in interaction with each other and with the context*" [p.4]. In this respect, *information* can then be further qualified as *content*, *meaning*, or *understanding*, where all these three concepts have their own specific properties. Therefore, speakers need *linguistic competence* to communicate, i.e what Saussure called *langue* [69] that is the items of knowledge which enable a speaker to make effective use of word-sign. This includes the use of the lexicon and syntactic structures. In this regard, this type of knowledge includes *semantic knowledge*, i.e., the knowledge about the literal meaning of words and sentences, as well as knowledge of facts and concepts, which helps in understanding the content and conveying information accurately; and *syntactic knowledge*, i.e., the knowledge pertaining to the rules and structures that govern sentence formation and grammar. This is what Coseriu [63] called *Linguistic Knowledge*.

As speakers of a specific language, there is also *Metalinguistic Knowledge* [70], that is the knowledge about language itself, including its structure, usage, and how it functions, enabling speakers to reflect on and discuss language explicitly. Beyond linguistic competence, which allows for the correct use of language, communication inherently depends on the context in which it occurs. We are here dealing with a set of information, such as *Contextual Information* that uses situational cues (e.g., time, place, social roles) to interpret meaning [4], *Domain Knowledge* that describes the information pertaining to a specific field, and *Discourse Knowledge* that involves understanding how utterances relate to each other within a larger conversation or discourse (e.g., *coherence* and *cohesion*) [310]. Additional skills are therefore necessary to interpret how language is used in different contexts and to achieve specific communicative objectives. In this respect, Paul Grice [125] describes communication as a *cooperative activity* between rational cognitive subjects. Those who engage in communication decide to do so because they share a (possibly minimal) goal they seek to achieve through cooperation, generating expectations. Those expectations correspond to *Conversational Maxims* (e.g. be truthful, relevant, informative, and perspicuous) that speakers follow during their cooperative activity [308] (see Chapter 2). This competence is referred to as *communicative competence* defined by Hymes [145] as the ability to discern when to speak or remain silent, what topics to address or avoid, and how to interact appropriately with others based on context, timing, location, and manner. Indeed, this *Pragmatic Knowledge* pertains to speaker performance and the contextual capacity to employ *parole* in communication [69]. Strictly connected to pragmatic knowledge, there is the ability to use and identify *speech acts* [13], namely the ability to understand the intentions behind utterances (e.g., requests, promises, suggestions). Thus, it can be stated that conversing is a linguistic task involving *understanding*, described by most pragmatists as an *inferential process* [125, 186], which pertains to an interpretive process that aims to uncover the meaning intended by a speaker. It uses the speaker's overt communicative behaviour and contextual information as input to produce a representation of the speaker's intentions and meaning [232]. The understanding of such intentions also relies on the identification of *implicatures*, i.e., implied meanings that go beyond literal interpretation [308], and *presuppositions*, i.e., assumptions made by speakers based on shared knowledge [286] (see Chapter 2).

In other words, the inferential capabilities of communicators are employed whenever it is necessary to retrieve implicit content. Let us analyse the following sentence taken from Minsky [211, 290]:

Jack needed money, so he went to shake his piggy bank. He was disappointed when he made no sound.

From this sentence, it can be inferred that Jack did not find any money, leading to a negative emotional response. This conclusion, while not explicitly stated in the passage, arises from the reader's *world knowledge*. Generally, a piggy bank is understood to be a container for coins, which are typically made of metal. Since metal is a hard solid, coins usually produce sound when shaken inside a container like a piggy bank; thus, the absence of sound indicates the absence of coins [290]. It is essential to recognise the role that *shared knowledge* - i.e. *knowledge constituted by all and only the things speakers both know* - plays as a premise for such inferences [308]. Such shared knowledge pertains to another set of information known as *Common Ground*, referring to mutual knowledge, beliefs, and assumptions [59] (see Chapter 2). Taking into account the example above, it can also be inferred that Jack is likely a child since this type of piggy bank is commonly possessed by children [290]. Alternatively, these predictions can be derived from similar events that individuals had in their childhoods, allowing them to draw similar conclusions based on analogy [211]. Indeed, as part of a society, speakers have *Socio-Cultural Knowledge* which refers to the awareness of social norms, cultural practices, and conventions that influence language use [129]. This knowledge helps in understanding appropriate ways to communicate based on cultural context and social expectations. While Socio-Cultural Knowledge is more structured and context-specific, CS is more general and universally applied. The two are related in that Socio-Cultural Knowledge can inform and enhance CS by providing a deeper contextual framework, helping individuals navigate and interpret everyday situations more effectively within a specific cultural context (see Chapter 3).

In the field of AI, therefore, there is a need both for communicative competence inherent to reasoning and also for sufficient coverage of the breadth of CS concepts for the retrieval of language understanding, perception, similarity and other cognitive functions [123]. In this respect, dialogue systems play a pivotal role, as they are

specifically designed to enable interactions that are safe [234], trustworthy [143], and personalised [268], addressing key aspects of human-like communication and cognitive engagement.

It is possible to distinguish two main types of dialogue systems: (i) Task-oriented Dialogue Systems (TOD) and (ii) Open-domain Dialogue Systems (ODD), serving distinct purposes in dialogue management. The TOD is specifically designed to efficiently manage task-oriented conversations, helping users achieve specific objectives by detecting user intentions, tracking dialogue states, executing appropriate actions, and providing relevant responses, involving a sequence of actions or sub-steps that are necessary to accomplish the main objective [241]. In contrast, the ODD is intended for open-domain interactions, facilitating free-flowing conversations across a wide array of topics by directly mapping the dialogue context to the response, without a predefined task or goal [143]. These two types address different use cases and user needs, demonstrating the versatility and applicability of dialogue systems in various scenarios [320]. Previous research has focused on the design and construction of these dialogue systems independently, using different structures to fit their distinct roles. As for TOD, these early systems operated within individual domains and adhered to predefined decision trees or rule sets. For example, GUS was limited to specific domains, such as flight information retrieval [36], while ALICE [317] was adopted to simulate a natural language conversation, following a script-like structure, based on predefined pattern matching and responses [161]. Despite their great performance in their application domains, they struggled to handle complex or nuanced conversations as they require extensive manual rule processing and lack the ability to generalise beyond their narrow domains. These limitations make them unsuitable for real-world applications with variable user intentions and open dialogues [320]. Nowadays, TOD has a broader spectrum of capabilities, embraces multi-dimensional functionality, and accommodates different user intentions, handling complex and multifaceted tasks within a single conversation [320]. This is possible due to the advent of the Neural Language Model (NLM), which uses the neural network as a probabilistic classifier to predict the likelihood of a sequence of words or to generate text [161] (e.g. BERT [71]), and subsequent advances in Recurrent Neural Networks (RNNs), consisting in a type of artificial neural network designed to recognise patterns in sequences of data, such as natural language [161] (e.g. Long Short-Term Memory, LSTM [138]), enabling more flexible and adaptive responses, allowing

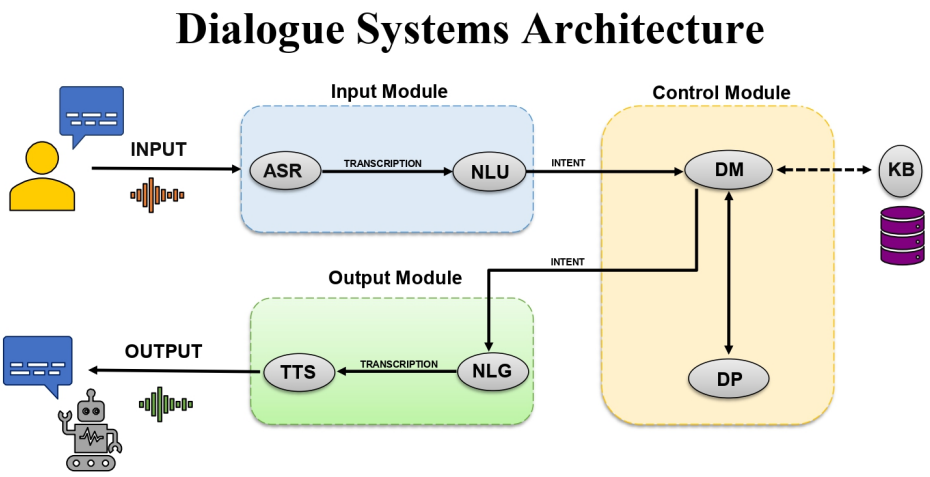

FIGURE 1.5: A general architectural example illustrating the language processing involved in an automated dialogue system, showcasing the flow of information from the user's input to the system's response. The architecture represented here is divided into three main macro modules: (i) Input Module, (ii) Control Module, and (iii) Output Module. In the (i) Input Module, the user's vocal input is processed by the ASR module, which transcribes the speech into text. The transcription is then analyzed in the NLU module to extract the user's intent. The intent is processed in the (ii) Control Module, where the DM module receives the intent from the NLU and may consult a KB to provide the DP module with the appropriate information to develop a suitable response to the user's request. Finally, in the (iii) Output Module, the NLG module creates the text of the response, which is then passed to the TTS to convert the text to speech and return the output to the user. It is important to note that the given example is specific for Spoken Dialogue Systems, as it contains ASR and TTS modules. For text-based Dialogue Systems, these modules are not included in the architecture and the user's input (e.g. a text) is processed directly in the NLU module and the output is generated directly from the NLG module.

systems to process and generate natural language in a context-aware manner [320]. This expanded capability was made possible by integrating six different components, which found similarities with the two aforementioned communicative models [73] (Figure 1.5):

1. **Automatic Speech Recogniser (ASR)**: In a spoken dialogue system, where interactions occur through spoken language, the speech input is handled by the Automatic Speech Recogniser (ASR). The ASR processes the input and provides a hypothetical transcription, along with a confidence score, by analysing acoustic patterns or using language models (grammars).

2. **Natural Language Understanding (NLU)**: The NLU processes user utterances to extract meaningful semantic frames through two key subtasks: (i)

intent classification and (ii) slot filling. It can operate using a rule-based approach, where predefined patterns or rules guide information extraction, or by leveraging deep learning techniques like intent classification and named entity recognition [249].

3. **Dialogue Manager (DM)**: The Dialogue Manager (DM) is responsible for mapping the abstract semantic or logical form of the speaker's input to the appropriate output, i.e., the response action that best matches the received input. It is the decision engine of the dialogue system, where action reflects the agent's level of understanding. In this decision engine, formal dialogue features are modelled either statistically or deterministically. These features may be sourced from external resources, such as by integrating domain representations in a graph database with probabilistic rules [307].

4. **Dialogue State Tracking (DST)**: The DST maintains and updates the conversation's status as it progresses. It keeps track of the user's goals, preferences, and any relevant context derived from the dialogue. The DST can either use a rule-based approach to update itself based on the NLU module's output or replace the NLU module entirely, directly tracking the dialogue status from the context using a similar architecture to the NLU [319, 17].

5. **Dialogue Policy (DP)**: The dialogue policy component determines the system's actions or responses based on the current state of the conversation. It translates the dialogue state into suitable actions, which might involve asking clarifying questions, offering recommendations, or making relevant API calls. Dialogue policies can be manually designed using rules or learned through methods like reinforcement learning [241] or supervised learning [136].

6. **Natural Language Generation (NLG)**: The NLG module produces natural language responses for the system based on the output of the dialogue policy. It converts structured information or system actions into clear and fluent sentences that users can easily understand. The NLG uses model-based methods, rule-based generation, or more advanced techniques like neural network-based generation models [338].

Since humans engage in natural language inference by drawing on a vast amount of external knowledge about both language and the world, and reasoning is a mechanism

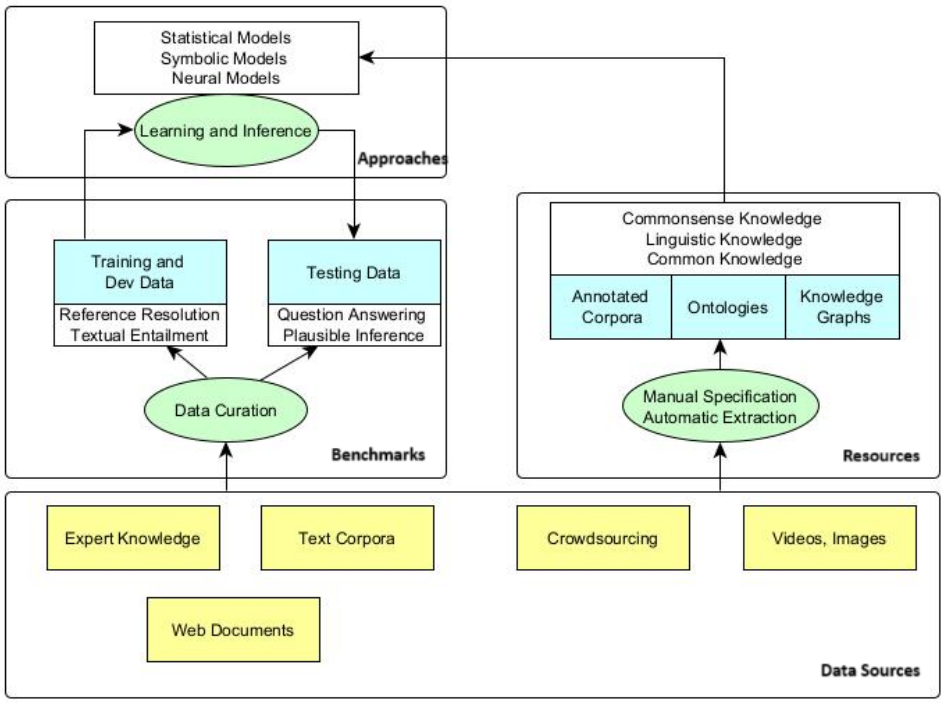

FIGURE 1.6: NLI research focuses on three key areas: knowledge resources, benchmarks and tasks, learning and inference methodologies. (Source: Storks et al., 2019 [290])

capable of generating answers to new questions by manipulating existing knowledge with inference techniques, representing knowledge in a suitable way is fundamental to improve conversational agents communicative performance [337]. In this regard, Knowledge Representation (KR) plays a pivotal role for both NLU and DM modules as it deals with the problem of *representing, maintaining and manipulating knowledge on an application domain* [172][p. 1], enabling an exhaustive and clear knowledge representation, fundamental for dialogue systems to solve complex tasks [220, 279, 154]. The NLP community has a rich history of developing knowledge resources containing CS information to support machines' inference ability [290]. In particular, the focus is on Natural Language Inference (NLI), defined as *the ability of machines to achieve a deep understanding of language that extends beyond explicit expressions, relying instead on conclusions drawn from general knowledge about the world* [290][p. 2]. Research efforts in NLI within the NLP community are primarily focused on three areas [290]: (i) knowledge resources (ii) benchmarks and tasks, and (iii) learning and inference methodologies, as summarised in Figure 1.6.

Information is often stored in Knowledge Bases (KB), i.e. repositories of knowledge that can be accessed and processed by systems [199] and represented in various

forms. Information can be represented as propositions, which are declarative statements that express ideas or assertions using logical forms, such as ATOMIC (Atlas of Machine Commonsense) [271], adopting an agent-centric perspective for representing events, focusing on abstract and social causes and consequences (e.g., *Person X leaves an object on the table and feels forgetful as a result*) [55]. Taxonomies consist of organising information in a hierarchical structure, such as WordNet [208], where entities are categorised into parent-child relationships to define classes and sub-classes. In semantic networks, such as ConceptNet [190], information is represented as a graph, where nodes represent concepts or entities, and edges represent their interrelationships. Ontologies are *a formal, explicit specification of a shared conceptualisation* [127, 128] of structural knowledge within a specific domain, consisting of classes, properties, and relationships. As cornerstones of the Semantic Web [33], ontologies have been essential for representing real-world knowledge as it represent entities in a set of triples (e.g. <*subject, predicate, object*>). An example of an ontology is OntoSenticNet [79], a commonsense ontology for sentiment analysis, containing the definition of precise conceptual hierarchy and properties associating concepts and sentiment values. Those repositories are built following three main methods [337, 290]: (i) *handcrafted methods*, involving a knowledge base manually constructed by human experts. An example is represented by Cyc [181], which was built by knowledge engineers who hand-code CSK into the CycL formalism. (ii) *Crowdsourcing methods* involve community members working together to build KBs, e.g. ConceptNet, which used a competitive online game to accept statements from humans in free text [281]; (iii) *text mining and information extraction methods* consist in automatically generate knowledge graphs and taxonomies from information sources on the Web, such as ATOMIC.

Since speakers refer to a broad spectrum of knowledge when communicating, in the literature this information is divided into three categories [290][p.4-5]:

1. **Linguistic Knowledge** refers to the basic skills of language comprehension (e.g. grammar) fundamental for any NLP system to understand natural language. Notable examples include annotated corpora such as the Penn Treebank [197], the first annotated corpus that guided the development of earlier machine learning approaches in the 1990s; lexical resources such as WordNet [208], which covers nouns, verbs, adjectives and commonly used adverbs and VerbNet [276], which is specifically focused on verbs, defining many verb

classes along with their argument structures, argument selection restrictions and syntactic descriptions. Semantic frame resources, such as FrameNet [16], provide a detailed information about verb semantic frames about prototypical events and situations (more details about FrameNet are provided in Chapter 2). Recently, there has been a growing adoption of pre-trained semantic vectors, employing numerical vector representations to capture word semantics, exemplified by word2vec [207], which train neural networks for word co-occurrence classification tasks generated from large texts;

2. **Common Knowledge (CK)** refers to well-known facts about the world that are often explicitly stated [49], therefore they can be mined from the Web with relative ease to create KBs. These include Wikipedia, the world's largest source of collaboratively edited encyclopedic knowledge, consisting of more than 2.5 million articles in over two hundred languages [315, 313] and from which other knowledge bases have been created, such as DBPedia [12], WikiTaxonomy [248], and Wikidata [3]. By extracting CK facts from Wikipedia, YAGO [292] augments WordNet information, converting it from a primarily linguistic resource to a common knowledge base;

3. **Commonsense Knowledge (CSK)**, which the capturing and encoding has been the subject of a long effort [68, 290] and several KB have been developed for it. These include Cyc [181], which contains rules expressing ontological relationships between objects (entities, collections, functions and truth functions) encoded in the CycL [182] language. ConceptNet [190, 284] is a semantic and multilingual network that describes concepts through words and their CS relationships. WebChild [299, 298] contains CS assertions that link nouns with adjectives through fine-grained relations such as *hasShape* and *hasTaste*. Rather than taxonomic or ontological knowledge as seen so far, ATOMIC contains inferential knowledge. It is a crowdsourced knowledge graph consisting of about 300,000 nodes corresponding to short textual descriptions of events and about 877,000 *if-event-then* triples representing nine types of *if-then relations* between everyday events.

Benchmark datasets are essential for assessing machines' progress on language processing tasks and facilitating the development of new approaches, helping to

[3]Wikidata: https://www.wikidata.org/

identify gaps in machines' capabilities and drawing insights from the complexities of human interaction to inspire more nuanced and effective language model [290]. Benchmarks can be designed in various formats, including multiple-choice questions that require binary decisions or selections from a list of candidates, as well as more open-ended questions [290]. Early research primarily focused on large-scale annotation collection to facilitate data-driven methodologies for low-level tasks, such as Part-Of-Speech (POS) tagging [197] and Named Entity Recognition (NER) [126]. In recent years, however, benchmarks have evolved to extend beyond linguistic context, taking into account implicit aspects of language to enhance understanding. These benchmarks exhibit considerable variation in terms of task scope. The Winograd Schema Challenge (WSC) [184, 183, 214, 217] focuses on specific reference resolution tasks, i.e. a linguistic mention in a span of text, that a particular expression refers to, such as a pronoun or phrase. Inspired by Winograd's work on computer systems for language understanding [325], this approach emphasises an integrated perspective that encompasses all aspects of language, including syntax, semantics, and inference. It is founded on the belief that effectively modelling language understanding requires addressing these elements in a cohesive manner [320]. In the WSC, systems are presented with questions about sentences (known as *Winograd schemas*). A classic example of a Winograd Schema is reported in the box below:

Winograd Schema Example

- The trophy did not fit in the brown suitcase because it was too big.

  What was too big?

To answer correctly to the question, a system must disambiguate a pronoun whose referent may be one of two entities and can be changed by replacing a single word in the sentence. In this context, linguistic constraints alone cannot determine whether *it* refers to the trophy or the brown suitcase. To interpret the pronoun *it* correctly, it is necessary to rely on external knowledge, e.g. the understanding that one object must be larger than another in order to contain it [184, 290]. Other benchmarks, such as The Recognizing Textual Entailment (RTE) Challenges [113, 30, 83] include broader tasks as textual entailment, defined as a directional relationship between a text and a hypothesis [65]. In this context, the text is said to entail the hypothesis if a typical person would reasonably infer that the hypothesis is true based on the information presented in the text. This task involves basic language processing skills,

such as named entity recognition and co-reference resolution. Since understanding what a typical person might infer is crucial, CSK is inherently essential for the textual entailment task. While textual entailment benchmarks necessitate drawing concrete conclusions, other benchmarks require making hypothetical, intermediate, or uncertain conclusions based on limited context [290]. This process is defined as plausible inference [68] and historically known as abductive reasoning [240, 137]. This reasoning process involves the use of linguistic context and external knowledge. Common tasks include cloze tasks, such as fill-in-the-blanks exercises, and sentence and story completion tasks [290]. An example of a Story Cloze Test task is represented by ROCStories [218]. This benchmark includes about 50,000 five-sentence stories from everyday life with causal and temporal relationships between events, making them ideal for learning CSK. Among them, about 3,700 stories are designated as test cases, each of which includes one plausible and one implausible alternative ending from which the trained systems must choose. A particularly important domain within plausible inference tasks is intuitive psychology, as the ability to infer emotions and intentions from behaviour is a fundamental human capability [122]. Consequently, benchmarks in this area necessitate intuitive social psychological knowledge, such as appropriate reactions in specific social contexts. To address this need, Rashkin developed the Story Commonsense benchmark [251], which includes around 160,000 annotations of character motivations and emotions within ROCStories, thereby facilitating more concrete reasoning in this domain. Additionally, systems may also need to infer intentions and reactions related to various events. To tackle this challenge, Event2Mind [251] is a dataset comprising approximately 57,000 annotations of intentions and reactions associated with about 25,000 unique events extracted from other corpora, including ROCStories. The diverse characteristics of these benchmarks serve different objectives, raising the critical question of which criteria should be considered to create benchmarks that effectively support technological advancement and provide measurable insights into research progress on deep reasoning abilities [290]. This is particularly relevant given that the capability to engage in natural interactions is fundamental for the implementation of human-machine communication [320].

Between 2019 and 2022 it has been assisted of the integration of some aspects of TOD and OOD, facilitated by advancements in deep learning and large-scale language datasets, leading to the emergence of Pre-trained Language Models (PTLMs),

models trained on vast amounts of unlabeled textual data, able to capture a rich tapestry of semantic and syntactic patterns [167, 250, 320]. An early attempt is represented by ELMo (Embeddings from Language Models) [243], a model able to capture context-dependent word representations [333]. With the developments of Transformers architecture [311], BERT model (Bidirectional Encoder Representations from Transformers) [71] was proposed, which pre-trains bidirectional language models using specially designed pre-training tasks on large-scale unlabeled corpora [333]. This paradigm produced a pre-trained dialogue model that subsequently undergoes fine-tuning using a dialogue corpus and the adopted backbone language models [191]. This technique has proven to be a groundbreaking solution to Neural Language Models (NLMs) [29, 206, 169], which needed extensive hand-crafted feature selection and domain-specific knowledge, making them cumbersome and time-consuming to develop for each task [320]. Researchers have observed a correlation between the size of the pre-training corpus and the model size with improved performance across various NLP tasks [162]. Therefore, it has been attempted to scale both the model size and the pre-training corpus simultaneously in order to enhance sample efficiency, enabling the models to learn more intricate patterns and representations from the data, transforming PLMs into Large Language Models (LLMs) [320]. The LLMs possess unprecedented capabilities across numerous language understanding and generation tasks, such as question-answering [244], thus transforming the landscape of dialogue systems. An example is represented by OpenAI's ChatGPT and GPT-4 [250, 311], from which a multitude of large language models have been proposed (e.g. LLaMa [306]). Corresponding dialogue models are subsequently fine-tuned on diverse dialogue corpora of varying sizes, to enhance performance across a range of dialogue tasks, domains, and even language [81], demonstrating a preliminary capability for both TOD and OOD, even in zero-shot or few-shot settings [331, 321]. These models are frequently depicted as repositories of CSK, acquired through extensive training on vast data collections that implicitly capture common knowledge and cultural patterns [294, 253, 142]. Unlike basic pre-programmed chatbots, LLMs operate as generative models possessing considerable inferential capacity and extensive reservoirs of knowledge, interacting with external contextual knowledge after deployment [235], e.g. user prompts, allowing them to dynamically adjust to the conversational context [191, 9]. Recent studies have shown that LLMs can parameterise factual knowledge during pre-training, which serves as an implicit knowledge base [244, 158, 297, 253].

1.2 Problem Statement

As seen so far, one of the aspirations of AI is to integrate conceptual and behavioural knowledge in machines to bridge the gap between humans and computers in problem-solving. Natural language, in particular, serves as a crucial medium and type of data, presenting numerous challenges for AI systems. The latest advances in this area promise to generate better actions and sequences to accomplish specific tasks, operating independently of external knowledge, practices, values, ideologies and similar constraints, suggesting the achievement of such outcomes even under conditions of greater uncertainty and randomness [274]. However, the programming of an intelligent system equipped with CS has so far eluded researchers. It can potentially be trained to operate within predefined parameters that correspond to CS expectations, but since CS is historically, culturally and socially bounded, it cannot be assumed that such programming is universal or objective, nor that it can be employed across the board in all markets [274]. In this regard, significant controversy remains over its multiple definitions, each of which emphasises different aspects of the concept [24]. To the general public, *Common Sense* is synonymous for *good judgment*. However, the AI community uses the term to refer to the vast number of fundamental facts and understandings that most people possess. Therefore, *Commonsense Knowledge* encompasses a significant portion of human experience, including understanding of spatial, physical, social, temporal, and psychological aspects of everyday life [6] (see Chapter 3). These factors are crucial considerations in the pursuit of developing conversational agents, as it is necessary to clearly define and conceptualise what are the main properties which constitute this knowledge (its *commonsensicality*) as well as its distinguishing features and the degree to which it is universally shared (its *commonness*). In a recent study conducted by Whiting & Watts [324], an analytical framework is proposed to quantify both of these elements. First, the commonsensicality of individual statements and people is characterised in terms of their tendency to agree on the former and their awareness of mutual agreement. Secondly, commonness is formalised as a clique detection problem on a bipartite belief graph of people and statements. Since it is a claim-based work, authors define CS as pq, i.e. the fraction q of statements shared by a fraction p of people. The framework is tested on a dataset of 2,046 evaluators who assessed 4,407 different statements, finding that CS aligns with simple statements resembling facts about concrete and everyday physical reality,

less with statements concerning subjective or abstract view of people and society. Furthermore, domain knowledge such as technology and the natural and physical sciences is often considered more commonsensical compared to other domains, such as philosophy, history and events, society and social sciences. Psychometric attributes such as social perceptiveness influence individual CS, while demographic factors, such as age or gender, do not have any influence. Collective CS is rare as only a small fraction p of people agree on more than a small fraction q of statements. As stated by the authors [324], these results pose a challenge to universalist beliefs about CS and raise questions about its variability.

In computational field, ontologies have long been recognised as effective means for facilitating communication and knowledge sharing between people and machines, providing a common understanding of a domain through a shared vocabulary (i.e., terms with unified meanings) [231]. They can be used as (i) a query model for data sources or as (ii) a basis for the integration of heterogeneous resources, e.g. Web pages, XML documents and relational databases, etc. [45]. In general, ontologies may cover the same or similar domains with different boundaries, perspectives/viewpoints, conceptual designs, granularity (i.e. levels of detail) and naming conventions. However, by presenting different information structures, problems of heterogeneity are inevitable [45]. Therefore, it is quite difficult to expect all organisations to agree on the use of a common ontology as developed independently of each other, in multiple places, by different communities and designers, for different applications and with different requirements, prerequisites and tools [45]. In this regard, the various approaches for building and integrating structurally different resources into a single database encounter distinct challenges related not only to data quality but also to limitations in the extensibility of these methodologies [337][p.284]: (i) the handcraft methods lack scalability, as evidenced by the limited size of knowledge bases such as CYC, which are constrained by the high costs associated with human labelling. (ii) Crowdsourcing methods often produce poor quality results and incorrect selections, as in ConceptNet, in which a significant portion of the assertions do not represent true CSK information as they are retrieved from existing knowledge bases, containing domain-specific information. Lastly, (iii) information extraction and text mining methods also have their shortcomings, containing a substantial amount of noise and biases as in WebChild. There are numerous tools available on the market for the

automated development of ontologies, such as Protégé [223], offering an intuitive interface for ontology editing and visualisation, project management, software engineering and other modelling tasks [283]. However, the fully automated derivation of ontologies from web sources without human supervision remains a time-consuming and labour-intensive research problem [171]. Overall, ontology-based systems, when considered in isolation, permit the representation (and reasoning) of only a portion of the entire spectrum of conceptual information [187]. This limitation arises not only from their inherent rigidity but also from their inability to adapt to dynamic domains, which further hinders their effectiveness [171].

The advancement of generative AI in recent years has been remarkably rapid as it holds the potential to accomplish specific tasks without reliance on external knowledge or experiences [275]. While current language model-based dialogue systems can convincingly pass the Turing Test and simulate human behaviour [236, 149], performing various tasks such as meal preparation or commuting to work, this advancement brings forth new problems and challenges that warrant further investigation [335, 320]. Despite these AI systems achieving nearly 90% accuracy in the Winograd challenge, effectively *defeating* it [168], this highlights the necessity for new benchmarks for more precise evaluation, as the ability to resolve anaphora is not equivalent of having *Common Sense* [274]. These models, which rely on the prediction of the next token during the pre-training phase, are adept at learning patterns from large amounts of static textual data [273] but still struggle to capture the subtlety of strategic reasoning. This limitation stems from the fact that strategic reasoning requires an understanding of complex and dynamic interactions between multiple agents, which cannot be deduced directly from static textual data alone, especially since large amounts of information are taken for granted, and thus not made explicit in texts [273]. In the study conducted by Ying et al. [329], the focus is on the robustness of LLMs in the case of conflicts. It is found that LLMs are highly susceptible to misleading cues, especially in the context of CS. Furthermore, in the study of Xu et al. [327] authors investigated how LLMs respond to knowledge conflicts during interactive sessions. The results suggest that they tend to support logically structured knowledge, even when it contradicts factual accuracy.

Based on these results, it is possible to state that AI remains fundamentally unaware of the tasks it performs and the rules that govern them, producing errors in critical thinking and hallucinations, significantly hindering their reliability in information-seeking tasks [155, 334, 188, 39, 18].

1.3 Research Framework Outline

The ongoing need for advancements in equipping AI with robust and adaptable CS capabilities and the obstacles represented by both generative AI and KBs are at the foundation of this work. Reasoning with uncommitted or unspecified logical forms poses significant challenges [265]. The primary obstacle these processes encounter is the availability of extensive CSK alongside a computationally efficient reasoning engine [265]. While the task of constructing comprehensive CSK repositories has been actively pursued by some researchers [181], many others have shifted away from the knowledge-intensive paradigm in favour of more quantitative methods [265]. In this regard, Charniak [54] empathises with the effectiveness of data-driven methods, without the need for large, manually constructed KBs. In contrast, researchers such as Rodney Brooks, a leading figure in the field of AI and robotics, argued that [it is] *better to use the world as its own model* [43] [p.140], thus rejecting the possibility of being able to create a database that could contain information capable of capturing all aspects of the human knowledge [265]. This perspective is principally articulated in Saba's work [265], where it is argued that limited progress has been made in recent years due to the fact that these systems primarily rely on symbol manipulation which lacks any substantive content. In particular, in these systems, there is almost no connection between semantics and our commonsense understanding of the world, which complicates the process of 'discovering' the significant amount of content that is implied, but almost never explicitly stated in our everyday discourse. Saba specifically argues that *a CSK structure should be discovered rather than created* [264][p. 610], highlighting that it is insufficient to merely establish a few principles for ontological design; rather, it is essential to develop a strategy for the systematic and objective design of an ontology of CSK. Moreover, the author argues that (i) understanding language largely involves CR at the pragmatic level and (ii) the framework of *understanding as reasoning*, together with the knowledge structures it employs, must be formalised to build scalable systems. This is in line with John

McCarthy's assertion, claiming that *we can hope to build intelligent systems when we can genuinely understand them* [200].

From this line of thought, it is therefore proposed in this work to frame this knowledge as a *process* rather than a static representation of facts. Given the vast amount of implicit information widely assumed by humans and the impossibility of representing it completely and exhaustively in a single resource, CSK is not necessarily confined to explicit representations of knowledge relations but can emerge through communicative interaction and be reconstructed through entities that present significant correlations with a high probability. In this regard, Verschueren et al. [312] argued that reasoning processes appeal to information from long-term memory, identifying two types of retrieved information: (i) frequency or likelihood estimations, and (ii) counterexamples, which pertain to a person's general knowledge about the world [47, 312]. This view fits well with Bayesian cognitive modelling, which has been the focus of many recent attempts to understand the interaction between structured representations and graded or statistical information [301, 121] (see Chapter 4). Since humans rely on various cognitive processes to make inferences and decisions based on prior knowledge, graph databases play a crucial role by offering a structured method to represent and navigate complex networks of information, reflecting the dynamic nature of human knowledge representation. In recent years, graph databases have gained popularity due to their ability to represent complex data in an interpretable and flexible format. They proved to be particularly useful in linking different resources, enabling the integrated analysis of information from multiple sources [228]. Therefore, data representation in the form of a graph, traced paths and the likelihood of finding specific types of relationships enable information extraction, reflecting the human approach to knowledge in a more dynamic way. In this study, Neo4j, a complete and self-contained graph database management system, has been employed as it is specifically designed to store, manage, and query data in graph form [323]. It allows one to define nodes and relationships between them, employing Cypher queries [105] to explore the graph structure for complex information retrieval [323] (see Chapter 4).

Since ordinary language is the best-known theory we have of everyday knowledge [265], developing a model that seeks to extract the processes underlying it directly from linguistic data seems to be more human-like. Therefore, a three-level analysis is

proposed here that aims to deepen the structure of this knowledge directly from the linguistic data. Drawing on the linguistic analyses, it will be possible to streamline the process of identifying CS information that can be omitted, which is in line with the principle of maximum utility for the design of a dialogue system.

Based on these assumptions, three research questions arise:

- **RQ1:** How does information get distributed in the dialogue flow?
- **RQ2:** Are there systematic relationships between actions and object categories?
- **RQ3:** Is it possible to identify linguistic features that distinguish implicit and explicit information?

To conduct the analysis, the cooking domain is taken into account for (i) its ubiquitous familiarity and (ii) the presence of implicit systematic chains of actions in the cooking process. In this regard, information is divided into three macro categories: Background, Foreground, and Presupposed Knowledge, each of which represents the depth of information (i.e., those explicitly mentioned in oral or written texts and those that are left implicit). The three levels of analysis focus on each of these knowledge typologies (Chapter 4 and Chapter 5). The first level focuses on the actions and entities involved in a surface situation (i.e., information explicitly formulated in a sentence). To answer **RQ1**, the dialogical corpus CookDial [157] is taken into account as it contains 260 user dialogues on the cooking domain, in which an agent provides the necessary instructions to a user for the completion of a recipe. To identify the semantic meanings of the explicit information, the FrameNet lexical base [16] was employed to identify the domain frames. For the corpus annotation, it was employed Label Studio [302], an open-source data labelling platform that facilitates the creation of annotated datasets, allowing the identification of frames distribution within the dialogic flow. The second level concerns the domain knowledge of entities and actions. In this regard, three datasets on the culinary domain are employed: (i) Recipe1M+ [198], which represents the ontology of recipes, instructions, and ingredients; (ii) FlavorDB [111], consisting of flavours ontology of ingredients; (iii) Epic-Kitchens [66] concerning routine actions performed in a kitchen (e.g. the use of ingredients, cooking appliances, tools and so on). These datasets are integrated

into a Neo4J database [323], an open-source graph database management system, representing the domain knowledge base. To answer **RQ2**, the frequency of Epic-Kitchens actions performed on ingredients belonging to specific categories within FlavorDB has been analysed. Lastly, the third level concerns the presupposed actions and entities, which allow the foreground information to be realised. Taking into account the Epic-Kitchens dataset, co-occurrence patterns of actions over entities have been extracted by applying a probabilistic estimation. To address **RQ3**, the linguistic characteristics of Epic Kitchens classes are taken into account to determine the distinction between implicit and explicit information. Based on the theoretical framework provided here and the results obtained from the analyses, an example application of managing kitchen work collaboratively is explained in Chapter 6 within Bastian, an embodied conversational agent built within FANTASIA [227], a plugin for the Unreal Engine designed to support the development of Embodied Conversational Agents.

CHAPTER

TWO

INTRODUCTION TO COGNITIVE LINGUISTICS

This chapter introduces key concepts in cognitive linguistics, with a particular emphasis on cognitive semantics. It will be explored the notions of semantic frame, cognitive frame, and communicative frame, each operating at the levels of language, thought, and communication, respectively [295]. The chapter then shifts focus to cognitive processes in communication, discussing the Theory of Mind, Grice's maxims, and presuppositions. These topics are essential as they provide the theoretical foundation upon which the framework presented in Chapter 4 is built, and a brief conclusion is provided at the end of the chapter.

2.1 Cognitive Linguistics

Cognitive Linguistics is interested in knowledge of the world and studies the question of how natural language contributes to it. Language is seen as a repository of world knowledge, a structured collection of meaningful categories that help to deal with new experiences and store information about old ones [112]. Cognitive linguistics is defined as a 'movement' or an 'enterprise', characterised by a diverse range of complementary, overlapping, and at times conflicting theories. This approach has adopted a common set of fundamental commitments and guiding principles from different disciplines, including cognitive psychology, rather than constituting a single, tightly articulated theory [90]. This approach initially emerged as a response to generative approaches to language. The Chomskyan generative tradition [57] emphasised the centrality of syntax, largely neglecting the importance of semantics and pragmatics in linguistic theory. This was disputed by many authors, highlighting the importance

of meaning. In this respect, Langacker [178] argued that:

"Meaning is what language is all about; the analyst who ignores it to concentrate solely on matters of form severely impoverishes the natural and necessary subject matter of the discipline and ultimately distorts the character of the phenomena described "[p. 12].

In this perspective, meaning arises from language use and is shaped by the activation of context-dependent conceptual knowledge structures. As a result, there is no clear-cut distinction between semantics and pragmatics [94, 88] as nearly no sentence conveys a complete thought on its own. Among the various aspects of the generative approach that were rejected was the idea of innate grammatical structures and language, especially through the concept of *universal grammar* [56]. The belief that linguistic knowledge was separate from other cognitive functions, suggesting the existence of a specialised brain module for processing language in isolation, was also completely rejected by these theories. From the beginning, cognitive linguists directly challenged these ideas, distancing themselves from these assumptions [20]. In this respect, cognitive linguistics focuses on explaining linguistic knowledge as a product of general cognitive abilities, stating that language arises from a general, usage- and frequency-sensitive learning mechanism, rather than relying exclusively on innate, domain-specific structures [46]. Moreover, cognitive linguists emphasise the importance of learning in language development, particularly in general socio-cognitive domain skills [303, 304]. Cognitive linguistics adopts an experiential perspective on conceptualization and meaning, observing that many of our concepts are rooted in our cultural and physical experiences. Specifically, it holds that our everyday embodied experiences play a crucial role in shaping our conceptual world [179]. As Gibbs suggests [115]:

"[...] Knowledge is seen by many cognitive semanticists as being grounded in patterns of bodily experience. These patterns, referred to as image schemas, arise through sensorimotor activity as we interact with objects, orient ourselves spatially and temporally, and direct our perceptual focus for diverse purposes" [p.233].

Research on the neural representation of concepts has primarily focused on the

role of sensory and motor information. This is due to the fact that conceptual representations of objects and actions develop largely through experiences involving perception and action [96]. Consequently, some theories suggest that comprehension of concrete words requires the reactivation of the sensory-motor representations that underpinned the acquisition of the corresponding concepts, as supported by Barsalou [22] and Glenberg & Gallese [116]. Several cognitive neuroscientific studies have provided important evidence supporting the grounding of knowledge in sensorimotor activity, aligning with the idea of image schemas. The main goal of cognitive neuroscience is "*to decode human brain activity — that is, to infer mental processes from observed patterns of whole-brain activation*" [259] [p.1]. To name a few examples, scholars such as Hauk et al. [133] demonstrate that conceptual processing is linked to sensorimotor brain areas. The authors argued that the organization of the motor and premotor cortex correlates with the somatotopic activation of these areas. This rules out a unified 'meaning center' in the human brain and supports a dynamic view in which words are processed by distributed neuronal assemblies with cortical topographies that reflect word semantics [133][p.301]. In 2016, Fernandino et al. [96] demonstrated how aspects of conceptual knowledge are encoded in multimodal and unimodal higher-level areas involved in processing the corresponding types of information during perception and action. Their results demonstrate a hierarchical system of abstracted sensorimotor representations, with a key division between object interaction and object perception processes. In this respect, Gallese and Lakoff [110] argue that the sensory-motor system has the right kind of structure to characterise both sensory-motor and more abstract concepts. Central to this picture are the neural theory of language and the theory of cognition, according to which, brain structures in the sensory-motor regions are employed to characterize the so-called 'abstract' concepts that constitute the meanings of grammatical constructions and general inference patterns. In this regard, the authors state [110]:

"[...] Our proposal is not an internalist theory of meaning. The reason is that imagination, like perceiving and doing, is embodied, that is, structured by our constant encounter and interaction with the world via our bodies and brains. The result is an interactionist theory of meaning. Accordingly, we argue that a key aspect of human cognition is neural exploitation — the adaptation of sensory-motor brain mechanisms to serve new roles in reason and language, while retaining their original functions as well." [p.456]

Lakoff [174] distinguishes two main commitments, which are fundamental guiding principles of cognitive linguistics [90]:

1. **Generalisation Commitment**. The focus is on identifying shared aspects of language and applying successful approaches and explanations from one domain to describe other linguistic phenomena. For example, taking the prototype model, several studies have applied the same principles to the organisation of morphology [300], syntax [117] and phonology [147]. In other words, the aim is to generalise successful explanations in the various domains of language for new purposes. In contrast to a horizontal perspective prevalent in other theories, this approach offers a more comprehensive, vertical view of language, providing additional explanations not available from a modular, horizontal standpoint [90].

2. **Cognitive Commitment**. This commitment emphasizes the importance of considering the cognitive basis of language, by integrating evidence from related cognitive and brain sciences [90]. It is argued that proposed models of language should reflect what is known about the human mind, rather than being driven solely by aesthetic preferences, such as the use of specific formalisms or economy of representation [64, 89].

Cognitive linguistics research can be broadly categorized into two principal domains [90]: (i) *cognitive semantics*, which focuses on modelling the human mind and examining linguistic semantics; and (ii) *cognitive (approaches to) grammar*, which concerns modelling the language system (the mental 'grammar') rather than the nature of cognition per se. It is important to note that although the study of cognitive semantics and cognitive approaches to grammar are occasionally distinct in practice, this does not imply that their areas of inquiry are anything but closely interrelated – most work in cognitive linguistics finds it necessary to investigate both lexical semantics and grammatical organization jointly [90]. For this reason, the terminology 'cognitive semantics' is employed here for simplification purposes.

2.2 Cognitive Semantics

Cognitive semantics is a branch of cognitive linguistics that examines the study of knowledge as it exists within the human mind, specifically the cognitive aspects of meaning. Semantics, traditionally understood as the theory of the relationship between language and the world [109], is reframed as cognitive semantics, which investigates how knowledge is represented (*conceptual structure*) and how meaning is constructed (*conceptualization*). Cognitive semanticists have employed language as a means to study these cognitive processes aiming to model the human mind as much as it seeks to understand linguistic semantics [90]. As Lemmens highlights [179][p.2], the term *cognitive semantics* can be misleading, as it suggests that semantics is a distinct module within the linguistic model, separate from 'cognitive syntax', 'cognitive morphology', 'cognitive pragmatics', and so on. As we have seen so far, cognitive linguistics does not adopt a modular view of language. Instead, it considers all linguistic structures, from morphemes to words to syntactic patterns, as inherently meaningful and of the same kind: symbolic form-meaning pairings, referred to as symbolic units or constructions. More specifically, grammar is defined as a structured inventory of such form-meaning pairs. Cognitive semantics follows four guiding principles, as described in Evans [90][p.6-12]: (i) **The *embodied cognition thesis*** posits that due to our physical attributes and neurological architecture, each of us have a distinct, species-specific perspective of the world. In other words, our interpretation of 'reality' is deeply shaped by the nature of our embodied existence. The fact that our experiences are embodied, structured by the characteristics of our bodies and neurological organization, has significant implications for cognition. Specifically, it pertains to the concepts we can access and the nature of the 'reality' we think and communicate about are a direct function of our embodied experiences. We can only discuss what we can perceive and conceive, and these perceptions and conceptions are fundamentally rooted in our embodied experiences [224]. (ii) ***Semantic structure***, which encompasses the conventional meanings associated with words and other linguistic units, can be equated with conceptual structure. Therefore, the meanings associated with linguistic units, such as words, represent only a fraction of the possible concepts. Humans possess a multitude of thoughts, ideas, and feelings that cannot be readily encoded in language. For instance, as Langacker posits [177], individuals have a concept for the region on the face below the nose and above

the mouth where moustaches grow. This conceptual understanding is necessary to understand that the hair in this area is termed a *moustache*. However, there is no conventional English word that encodes this specific concept. Consequently, the set of lexical concepts, the semantic units conventionally associated with linguistic units like words, comprises only a subset of the full range of concepts present in the minds of speakers and listeners [87, 88, 91]. (iii) ***Meaning representation is encyclopedic***, which means that lexical concepts do not represent packaged bundles of meaning - in contrast to the *dictionary view* supported by Haiman [131]. Rather, they serve as 'points of access' to vast repositories of knowledge relating to a particular concept or conceptual domain [177]. The conventional meaning associated with a particular linguistic unit serves as a 'prompt' that drives the process of meaning construction: the selection of an appropriate interpretation based on the context of the utterance (see Chapter 4). Lastly, the fourth guiding principle states that (iv) ***meaning construction is conceptualization***. Language itself does not encode meaning, rather, words serve as *cues* that guide the construction of meaning. Consequently, meaning is constructed at the conceptual level. This process of constructing meaning is equated with conceptualization, where linguistic units act as cues for a series of conceptual operations and the assumption of background knowledge. Ultimately, **meaning is a process**, not a discrete 'thing' that can be 'packaged' by language [90] (see Chapter 4).

As we have seen so far, cognitive semantics defines meaning as conceptualization. Semantic structure is understood as conceptualization tailored to linguistic convention, requiring the explicit characterization of conceptual structure [177][p.99]. While the cognitive view may initially appear similar to theories considering meaning as concepts or conceptual representations, it differs in significant ways. These include its encyclopedic perspective on meaning and its stance that meaning is non-truth-conditional. The conceptual structure underpinning linguistic expressions can range from simple concepts or perceptual experiences to complex knowledge clusters, such as Fillmore's frames, i.e. *unified frameworks of knowledge, or coherent schematizations of experience* [99][p.223].

2.2.1 Semantic Frames

The concept of *frames* in Linguistics emerged from Bartlett's schema theory [23] in Cognitive Psychology, and is commonly attributed to Fillmore's work on *Frame*

Semantics [101]. The word *frame* itself was introduced into Linguistics in the form of *syntactic frames* by Fries [107], which typically consists of sentences with a blank space representing a missing word. Specific words can then be inserted into the gap to test their acceptability within that syntactic frame [295]. Inspired by this theory, Fillmore proposed the concept of *case frames*, i.e. sentences with a designated space that can only be accommodated by words with a specific semantic function, such as Agent, Instrument, and others [98][p.46]. Case frames [97] were conceived of as:

"[...] characterizing a small abstract 'scene' or 'situation', such that comprehending the semantic structure of a verb required understanding the properties of these schematized scenes" [p.115].

While Fillmore initially identified six cases, he acknowledged the likelihood of additional cases being required as the framework expanded. The evolution from case frames to semantic and subsequently cognitive frames (which a definition will be provided in the next paragraph) was facilitated by the work of Minksy [212], who employed the term *frame* to denote *a data-structure representing a stereotyped situation* [p.212]. In a concise manner, Minsky proposes that frames can model a diverse array of human capacities, spanning from visual perception to grammaticality judgments [295]. Thus, the *Frame Semantics* research program emphasizes the continuities of relationships between language and experience.

A *frame* refers to an interconnected system of concepts, where understanding one concept requires grasping the entire system; introducing any one concept makes the whole system accessible. In this context, a word represents a category of experience, and part of the research is uncovering the reasons a speech community has for creating and defining the meaning of that word category [245][p.1-2]. The notion can be exemplified through the *Commercial Transaction Frame* [98, 295]. This frame encompasses a typical actors and entities involved in a commercial transaction, such as a *buyer*, a *seller*, *goods*, and *money* and it is associated with a large set of semantically related verbs, including *buy*, *sell*, *pay*, *spend*, *cost*, and *charge*, each of which highlights different aspects of the frame. For instance, the verb *buy* emphasizes the buyer and the goods, while downplaying the seller and the money; *sell* focuses on the seller and the goods, de-emphasizing the buyer and the money. Lastly, *pay*

centers on the buyer, the money, and the seller, with the goods being backgrounded. The underlying premise is that understanding the meaning of any one of these verbs necessitates comprehending the dynamics of a commercial transaction and grasping the meaning of any single verb implies, to some degree, understanding the meaning of all of them. The knowledge and experience structured by the Commercial Transaction Frame serve as the foundation and motivation for the categories represented by the words. In other words, the linguistic elements evoke the frame, which the interpreter then invokes [245][p.2-3]. Semantic frames give birth to the FrameNet project [16], which is narrowly focused on the semantic prerequisites of lexical items. A frame is characterized as *a script-like conceptual structure that portrays a specific type of situation, object, or event together with its participants and props* [261][p.7]. The 'participants and props' are *frame elements* composed of frame roles and their corresponding fillers [295]. This lexical base was employed in this work for semantic annotation and more information about it will be provided in Chapter 5.

2.2.2 Cognitive Frames

Semantic frames do not exist in isolation as they depend on *cognitive frames* explicitly recognized in Fillmore's later work [7, 258, 100]. The ability to think is a prerequisite for language and communication, and cognitive frames must exist for semantic frames or communicative frames to be possible. They represent the most well-known type of frames in the field of linguistics, as they are featured prominently in Lakoff's research [176] and some of Fillmore's landmark publications [97], even though Fillmore did not use the specific term "cognitive frames" until a later point in time [295]. Fillmore [97] defined a *frame* of this kind as:

"[...] any interconnected system of concepts wherein understanding any single element requires comprehending the entire structural context in which it is situated; when one component within such a structure is introduced in a text or conversation, the other related elements are automatically made accessible" [p.111].

As highlighted in Sullivan [295], Minsky [212] describes a type of frame akin to Fillmore's cognitive frames, using the example of a **BIRTHDAY PARTY** frame. This frame structure encompasses expectations such as gifts, cake, candles, and so on. In this respect, we can recognize a birthday party event without verbal or other

forms of communication, indicating that the frame is cognitive rather than semantic or communicative in nature [295]. Other concepts such as *birthday presents* and *cake* are evoked. For instance, if we are invited to a birthday party, we may consider purchasing and wrapping a gift, bringing a card, dressing nicely, and congratulating the person celebrating their birthday. These are all elements of the cognitive frame of **BIRTHDAY PARTY**. However, a mere list of these elements would not be enough to explain a birthday party to someone from a different cultural and linguistic background [295][p.8-9]: they would need to understand the *solar calendar* and the concept of *tracking age in years*, as well as the notion of *material property* and *exchange* that allows gifts to be bought and given. These additional requirements are part of the **BIRTHDAY PARTY** cognitive frame but are inherited from more basic frames such as **SOLAR CALENDAR**, **FINANCIAL TRANSACTION**, and **GIVING**. We can think about the solar calendar without considering birthdays or parties, but the reverse is not true - **SOLAR CALENDAR** is inherited by **BIRTHDAY PARTY**, but not the other way around. Understanding a high-level frame like **BIRTHDAY PARTY** requires grasping many fundamental frames [295]. In this respect, Fillmore [97] states that:

"[...] often the frame or background against which the meaning of a word is defined and understood is a fairly large slice of the surrounding culture, and this background understanding is best understood as a 'prototype' rather than as a genuine body of assumptions about what the world is like" [p.117–118].

This means that it is necessary to know what the typical birthday party is like to understand the **BIRTHDAY PARTY** frame [295]. In this context, the *Prototype theory* [256, 32] plays an important role in cognitive frames as even if we do not expect that every birthday party to match the prototype, we are still aware of it and can refer to it [295]. From this perspective, the frame is not just linguistic knowledge, as it influences how people think and act even in the absence of language (e.g. not verbalised). People who is familiar with the **BIRTHDAY PARTY** frame can recognize a birthday celebration without the explicit use of words that refers to it, such as *birthday*, *balloons*, or *cake* [295]. Therefore, cognitive frames represent a form of *encyclopedic knowledge* [165] as words like *birthday* provide access to a vast amount of information structured by these cognitive frames. This way, the communication is

facilitated by allowing people to reference shared conceptual structures, even when they are not actively speaking or communicating, such as when contemplating the time of year or planning a child's birthday event [295]. It turns out that cognitive frames are essential to language, as speakers depend on them to understand words and are necessary background knowledge for semantic frames.

Fillmore explains the relationship between cognitive frames and semantic frames, defining the cognitive frames as *those background understandings needed for making sense of things that happen around us*, which are activated by a wide range of thoughts, experiences, and language. On the other hand, semantic frames are *those that are specifically coded in—or "evoked by"—lexical units or other features of linguistic form* and are evoked only by units of language [7] [p. 158], [295].

2.3 Cognitive processes in Communication

The main goal in conversation analysis is to recognise what the interlocutors know and be able to adjust one's actions and understandings accordingly [73]. These abilities are directly linked to the *Theory of Mind* (ToM), consisting in the ability to infer and predict the intentions, thoughts, desires, and behavioural reactions of oneself and others, through an awareness that others have a mind with "affective" and "cognitive" mental states that may differ from one's own. This understanding shapes how utterances are interpreted and the contexts in which they are spoken, becoming an essential prerequisite for successful human social interaction [1]. The false belief task, as described by Ruffman [260], is a method for assessing the ToM. This task involves testing a child's ability to comprehend the conflict between a presupposition and new contextual evidence that has not yet become part of another person's shared understanding. This task involves a character (Maxi) who places some chocolate in a particular location and then leaves the room; while he is gone, the chocolate is moved to another spot. The child is then asked where Maxi will look for the chocolate upon his return. To succeed in this task, the child must understand that Maxi still believes the chocolate is where he left it, i.e. the child must recognize that the person has a false belief [260][p.2]. The inference of mental states corresponding to propositional attitudes guides interaction and can shape the use of specific conversational feedback

[73]. In other words, inferring another agent's mental state is a prerequisite for depicting the *communicative frame*.

2.3.1 Communicative Frames

Semantic frames and cognitive frames contribute to *communicative frames*, which Entman [85] defines as:

" The process of selecting and emphasising certain aspects of a perceived reality within a communicating text. This frame serves to promote a particular problem definition, causal interpretation, moral evaluation, and/or treatment recommendation for the item described" [p.52].

Therefore, frames are the means by which information is presented and understood, which can subsequently influence how individuals interpret and respond to that information [52]. Many of our thoughts can be communicated through language, art, music, or other means and when we convey a cognitive frame to others, they may activate that frame. The relevant structure of the cognitive frame selected for communication, along with the factors that influence whether and how the frame is communicated, constitute a communicative frame. The structure of one or more cognitive frames is therefore encompassed within each communicative frame, just as each semantic frame includes only a subset of the structure in a cognitive frame [295][p.9]. Even if Fillmore never explicitly stated in his work, Sullivan [295] highlighted that the author implied the existence of communicative frames, contending that *the same 'facts' can be presented within different framings, framings which make them out as different 'facts'* [97][p.125]. For instance, the words *thrifty* and *stingy* describe the same behaviour of avoiding spending money, but they make different assumptions about the value of that behaviour. *Thrifty* assumes that saving money is commendable and waste is undesirable, while *stingy* suggests that spending money can be beneficial and unwillingness to do so can be seen negatively. To reject the framing imposed by *stingy* and the assumption that spending is advantageous, one must select a word with different connotations, like *thrifty*. Fillmore's concept of *different framings* thus falls under the category of communicative frames [295][p.10].

2.3.2 Conversational Maximes

The inference of mental states corresponding to propositional attitudes guides interaction and can shape the use of specific conversational feedback [73]. In dealing with mental processes involved in communication, cognitive semantics shares basic properties with pragmatics, as language is analysed not as an abstract structure but as a human quality [90]. Indeed, cognitive pragmatics aims at investigating what happens in participants' minds [144][p. 281]. In Grice's account [125] communication is described as a cooperative activity between rational cognitive subjects [308]. Individuals who engage in communication decide to do so because they share a goal that they aim to achieve through cooperation, generating expectations. Grice clarifies the types of expectations which communicators are subject to by enunciating a principle that should govern their cooperative activity [125][p.45]. This principle is known as **Principle of Cooperation**, which states:

Make your conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged.

The cooperative model of communication proposed by Tomasello [304] posits that the communicator (C) has individual goals (e.g. personal objectives and values), and when C perceives the recipient (R) can assist in achieving these aims, C will take specific actions intended to motivate R to do, know, or share something. This reflects C's social intention. If R comprehends C's social intention, R can then decide whether to cooperate as expected. This cooperative dynamic is manifested through the communicative act, whether verbal or non-verbal, which is mutually represented within a shared attentional frame. Consequently, C's communicative intention becomes shared as well. Additionally, C may direct R's attention to a referential situation in the external world to guide R to infer social intentions through processes of cooperative reasoning [144][p.282]. Conversely, R first attempts to identify the referent, typically within the shared knowledge, and then to infer the social intention, also by relating it to that shared knowledge [304, 144]. From it, Grice formulates a series of maxims that specify what the audience should expect from the speaker, as reported below [308][p.72]:

1. **Quantity**: (i) *Make your contribution as informative as is required*; (ii) *Do not make your contribution more informative than is required*. These maxims articulate the expectation that the speaker should be sufficiently informative but not excessively so. The best way to be cooperative is to provide just the right amount of information required by the purposes of the conversation

2. **Quality**: (i) *Do not say what you believe to be false*; (ii) *Do not say that for which you lack adequate evidence*. It specify that the speaker should provide only information that is reasonably true. The idea is that giving false information does not help the audience.

3. **Relation**: (i) *Be relevant*. This maxim highlights the importance of the relevance of a speaker's contribution to the conversation.

4. **Manner**: (i) *Avoid obscurity of expression*; (ii) *Avoid ambiguity*; (iii) *Be brief*; (iv) *Be orderly*. These maxims articulate the expectation that the speaker should be clear.

By following the conversational maxims, a speaker facilitates the interpretation process for the audience. If the audience recognizes the speaker's communicative intention, they may expect the speaker to be cooperative and follow these maxims [308]. In this respect, humans demonstrate the capability and willingness to share mental states [144][p.283]. Among these mental states, beliefs can be categorized into three types: (i) *individual*, i.e. personal beliefs that agents hold for themselves or in representing others, without an existing connection between the agents; (ii) *common*, i.e. individual beliefs that agents mutually share in a given context; (iii) *shared*, i.e. beliefs that are common to all participants engaged in a conversation, and that each participant knows are possessed by all other participants. Cooperation and intention help identify the sharedness of mental states [74]. This process is important in this work as it includes a set of information such as *Commonsense Knowledge* and *Common Ground*, which will be explored in Chapter 3. In this respect, Grice [125] makes a distinction within speaker meaning between what is explicitly stated and what is implicated. What is explicitly stated is mainly interpreted by means of the knowledge of the rules of language and contextual information. Explicit communication can therefore work well within the compliance with conversational maxims; in contrast, implicit content must be somehow inferred by the audience [308]. These

types of knowledge, especially as far as Commonsense Knowledge is concerned, are accessed through the inferential process [125, 186], which is an interpretive process that aims to discover the meaning intended by a speaker. It uses the speaker's overt communicative behaviour and contextual information as inputs to produce a representation of the speaker's intentions and meaning [232]. The understanding of such intentions relies on the identification of *implications*, i.e., implicit meanings that go beyond literal interpretation [125, 308], and *presuppositions*, i.e., assumptions made by speakers on the basis of shared knowledge [286]. In this work, the focus is exclusively on presuppositions, exploring their role and importance in the communicative exchange.

2.3.3 Presuppositions

The notion of presupposition entails defining the concepts of truth, falsity, and logical form [233]. Early discussions on presupposition largely centered around definite descriptions, which are claimed to presuppose the existence of a unique referent [26]. A problem emerges when a definite description, such as "The King of France", fails to refer. Russell [263] asserted that sentences like *The King of France is bald* are false because the logical form of definite descriptions contains a false existential claim. Conversely, Strawson [291] (known as *Frege-Strawson approach*) famously countered Russell's theory by proposing that when a definite description lacks a referent, the result can be a sentence devoid of a truth value. Thus, presuppositions are understood as definedness conditions, necessary prerequisites for an expression to have meaning [26]. The controversy of Strawson and Russel over *The King of France is bald* represents the beginning of the long-standing controversy over the nature of presupposition [233][p.58]. By the early 1970s, more linguistically-oriented research had expanded the empirical domain of presupposition theory, as discussions of presupposition also engage with the underlying theory of meaning related to, yet extending beyond, logical form [233]. In this regard, the conceptualization of pragmatic presupposition, which underscores the speaker's considerations, the context of the utterance, and the knowledge shared by the conversational participants that are taken for granted is highlighted [233, 27]. The most significant philosophical counterpoint to the Frege-Strawson approach to presupposition is attributable to Stalnaker [286], which argues that:

"A proposition P is a pragmatic presupposition of a speaker in a given context just in case the speaker assumes or believes that P, assumes or believes that his addressee assumes or believes that P, and assumes or believes that his addressee recognizes that he is making these assumptions, or has these beliefs" [p.473].

Therefore, stating a sentence whose presupposition is known to be false would simply result in an infelicitous utterance, rather than a truth-valueless sentence as semanticists suggest [233]. In this respect, semantic presupposition is a relation between sentences, defined in terms of *entailment*. Entailment is a relation between two sentences where the truth of the second sentence necessarily follows from the first; one cannot assert the truth of one sentence and deny the truth of the other. Negation is a useful test to distinguish entailment and presupposition [166, 185]. While background knowledge is considered essential for interpreting utterances, the verbal discourse itself and the context of the utterance are equally crucial. Therefore, background knowledge should be considered alongside the semantic representation of a sentence spoken in a particular context [233]. Specifically, in sentence interpretation, the principle of pragmatic presupposition ought to complement, rather than substitute, the semantic representation. The task of pragmatics is to explicate the relationship between how utterances are understood and the contexts in which they are spoken [185][p.1-31]. In this respect, various scholars [11, 166] have presented detailed arguments that presuppositions should be understood as akin to conversational implicatures, justifying presuppositional inferences by invoking the maxims of relevance and quantity [26].

2.4 Conclusions

This chapter has addressed issues pertinent to cognitive semantics, a branch of cognitive linguistics that examines the study of knowledge as it exists within the human mind, specifically the cognitive aspects of meaning. Cognitive semantics defines meaning as conceptualization. Semantic structure is understood as conceptualization tailored to linguistic convention, requiring the explicit characterization of conceptual structure. The conceptual structure underpinning linguistic expressions can range from simple concepts or perceptual experiences to complex knowledge clusters, such as Fillmore's frames, emphasizing the continuities of relationships between language and experience. A *frame* refers to an interconnected system of concepts,

where understanding one concept requires grasping the entire system; introducing any one concept makes the whole system accessible. Semantic frames do not exist in isolation as they depend on *cognitive frames*, defined as an interconnected system of concepts wherein understanding any single element requires comprehending the entire structural context in which it is situated; when one component within such a structure is introduced in a text or conversation, the other related elements are automatically made accessible. Semantic frames and cognitive frames contribute to *communicative frames*, which Entman defines as *the process of selecting and emphasising certain aspects of a perceived reality within a communicating text. The ability to think is a prerequisite for language and communication, and cognitive frames must exist for semantic frames or communicative frames to be possible*. In this respect, in conversation analysis, the main aim is to recognize what the interlocutors know and be able to adjust one's actions and understandings accordingly. These abilities are directly linked to the *Theory Of Mind*, which involves the capacity to attribute knowledge, beliefs, and intentions to others, facilitating more effective and coherent communication. In dealing with mental processes involved in communication, cognitive semantics shares basic properties with pragmatics, as language is analyzed not as an abstract structure but as a human quality, aiming at investigating what happens in participants' minds. In Grice's account, communication is described as a cooperative activity between rational cognitive subjects who engage in communication as they share a goal that they aim to achieve through cooperation, by enunciating a principle that should govern their cooperative activity. From it, Grice formulates a series of maxims that specify what the audience should expect from the speaker, which are known as Conversational Maxims. Through this process, humans demonstrate the capability and willingness to share mental states, which can be categorized into three types: (i) *individual*, (ii) *common*, and (iii) *shared* beliefs. Cooperation and intention help identify the sharedness process of mental states, which includes a set of information such as *Commonsense Knowledge* and *Common Ground*, which will be explored in Chapter 3. In this respect, Grice makes a distinction within speaker meaning between what is explicitly stated and what is implicated. What is explicitly stated is mainly interpreted by means of the knowledge of the rules of language and contextual information, where explicit communication can work well within the compliance with conversational maxims. In contrast, implicit content must be somehow

inferred by the audience. These types of knowledge, especially as far as Commonsense Knowledge is concerned, are accessed through the inferential process, which is an interpretive process that aims to discover the meaning intended by a speaker. More specifically, the understanding of such intentions relies on the identification of *presuppositions*, i.e., assumptions made by speakers on the basis of shared knowledge. Therefore, background knowledge should be considered in conjunction with the semantic representation of a sentence spoken in a particular context, as the principle of pragmatic presupposition enhances, rather than replaces, the semantic representation. As we will see in the next chapter, Common Sense is essential for effective communication, which is accomplished by a deep understanding of language and the world, becoming crucial for constructing relevant, coherent, and engaging responses within an ongoing conversation [102].

CHAPTER

THREE

UNRAVELLING COMMON SENSE

The chapter explores the concept of *Common Sense*, a term that has been studied and analysed in various fields throughout history. What emerges from the work conducted by Bauer & Schiele [25], the development of AI appears to lie at the intersection of the 18th-century Scottish/English Enlightenment philosophy and Vico's proposal, conceiving Common Sense (CS) as an umbrella term that encompasses *Commonsense Knowledge* (CSK) and *Commonsense Reasoning* (CSR), a *communal sensibility* that unites people through shared attention and intentionality. While avoiding an exhaustive treatment of this philosophical background, I will outline the general lines of thought to provide a foundation for understanding CS and its connections to what has been said in cognitive linguistics.

3.1 Definition of Common Sense

In the Collins English Dictionary *Common Sense* (CS) is defined as "[a] natural ability to make good judgments and to behave in a practical and **sensible** way" [1]. The term is widely invoked across diverse fields, from philosophical discourse [180], history [257], sociology [140, 322], psychology [150] to the assessment of AI [68]. Its diverse applications help capture every possible nuance from diverse perspectives, contributing to a more comprehensive and multidimensional definition of the concept [324].

[1]Collins English Dictionary: https://www.collinsdictionary.com/dictionary [last visited: 26/09/2024]

In Lewis' work *Studies of Words* [186], the author distinguishes four uses of the term '**the** *Common Sense*' (with definite article) [pp.146-147]:

1. The CS refers to a faculty common among human beings that describes the 'elementary mental equipment of a normal man';

2. The idea of social virtue pertains to a quality that includes courtesy and the feeling of friendship acquired in interaction with other human beings. This is a secondary benefit of public schools, as opposed to public schools, where children acquire a sense of community in relationships with their peers;

3. The CS also refers to the common experience of human beings, to their pains and pleasures, to the emotions of gain and loss, to birth and death, to the comic and what is therefore praiseworthy. These are the commonplaces to which public speakers can universally appeal;

4. The CS is a technical term from ancient and medieval psychology, referring to internal senses such as memory, imagination and common sense; the latter is the arbiter of the others.

In these points, Lewis consolidates all possible interpretations, which are then declined differently in each field of study. Another key aspect highlighted by Lewis is the dual connotation of the term. Indeed, CS can be used in a derogatory sense and to emphasize a universal potential shared by all humans [274]. In this respect, there is a long-standing tradition of philosophical disputes that devalues CS, often to the detriment of scientific inquiry. The genealogy of the denigration of CS is reconstructed in Waldenfels' work [314], starting from Greek philosophy in terms of the opposition between Doxa (i.e. mere opinion) and Episteme (i.e. solid knowledge), demarcating the difference between CS knowledge, which consists in everyday thought, lay knowledge, and opinion, and scientific knowledge, which is an expert, specialised, evidential and deductive knowledge [25]. This dispute between Doxa and Episteme can be understood through distinct perspectives. Some scholars, such as Wolpert [326], assert that the two are fundamentally different, as CS represents flawed and misleading cognition, while science is counterintuitive but avoids the pitfalls of natural language by employing rigorous quantitative concepts and methods. Furthermore, it is argued that scientific evidence must ultimately replace CS to improve the world. In these terms, wherever CS prevailed, scientific knowledge must

take precedence [274]. In contrast, other academics, such as Bronowski [42] and Hoyningen-Huene [141], support the *continuity hypothesis*, which posits that science is a systematic extension of CS, a little more elaborate, refined and precise rather than a complete departure from it [25]. To this long tradition of vertical or horizontal ordering and denigration of CS and related concepts such as Doxa, everyday thinking, secular knowledge, and opinions, it is crucial to examine the parallel history of the deployment of different conceptions of the term itself, which primarily draws on three historical-philosophical strands [25][p.16-21]:

1. **Aristotle's (384-32 BC) sixth sense as *koine aisthesis***, where *Koine* refers to *common* and *aisthesis* refers to *sensation*, translated as *common sensitivity*. The conception of **the** CS encapsulates the 'inner perception', i.e. the ability of *seeing that we see and hear*, accompanied by the monitoring (but not direct observation) of perceptual activity, and the retrospective accessibility of mental states in second-order consciousness [25].

2. **The Scottish/English Enlightenment (18th century)**, with Thomas Reid and G.E. Moore as its leading exponents [252, 213], CS (without the definite article) refers to accessible knowledge, i.e. the stock and flow of inputs processed by individuals, requiring no expertise beyond being born into a society and undergoing primary socialization as a part of the human experience. To retrieve this knowledge, it is necessary to employ a mode of reasoning which is between intuitive and formal-deductive rationality [132].

3. **Giambattista Vico's Philosophy (1668-1744)** [247], where CS emphasizes social integration. Here, CS is the human capacity to orient toward others for the mutual common good, where the *moral community* is not the result of abstract reasoning but rather of joint attention and intentionality focused on common problems. CS becomes reflexive as the historical progress of a society with communicative rationality, in speech acts of deliberation, reasoning and collective sense-making.

From these strands, three key concepts can be extracted in delineating the CS definition [25][p.14] [2]:

1. **CS solves the "binding problem", i.e., integration of multi-modal sensory experiences**. This ties into the experientialist approach to cognitive semantics, as articulated by Johnson [160] and Lakoff [175]. According to this perspective, experience is seen in a broad sense, encompassing sensory-motor, emotional, social, and other forms of experience shared by all normal human beings, including innate capacities that shape and enable such experiences [173]. In this context, the term *experience* does not refer to individual, unique experiences but to the common human experiences grounded in our bodies and the real-world context in which we function. These shared experiences actively engage us with the world, enabling conceptual understanding and reasoning in a way that is central to human cognition. Though experience does not strictly determine thought, it motivates and shapes the conceptual structures and reasoning processes characteristic of human thought [160].

2. **CS is a stock of universally recognisable knowledge to reason from and is the quasi-rational judgement call: neither pure intuition nor formally rational**. Concerning the conception expressed by the Enlightenment philosophy, it can be stated that CS serves as an umbrella term to encompass both *Commonsense Knowledge (CSK)*, i.e stock of knowledge comprising a large number of facts and beliefs about the world which is widely shared and accessible among people [48, 257] and *Commonsense Reasoning (CR)*, i.e. the process of gathering information about specific elements of a scenario and drawing inferences about other aspects based on CSK or an understanding of how the world works [219]. In this respect, the development of AI attempts to load the knowledge base into the machine with the construction of databases and systems that emulate the process of reasoning, as seen in Chapter 1. From a cognitivistic perspective, CSK corresponds to cognitive frames, a shared conceptual structures of concepts not actively speaking or communicating in which understanding any single element necessitates comprehending the entire structural context in which it is situated. Likewise, CSR corresponds to

[2] In Bauer's work, the author outlined eight key concepts that define the concept of CS. For this work, I reported here only three of them as they most closely match my theoretical approach.

the presupposition process, whose task is to explicate the relationship between how utterances are understood and the contexts in which they are spoken.

3. Lastly, **CS is the communal sensitivity that binds us into joint attention and intentionality**. In Vico's view, CS is tied to the community of speakers and listeners engaged in conversation and dialogue [25]. His conception of CS pertains to a widespread belief, opinion, or *common ground* that can be invoked in discourse. This knowledge does not need to be made explicit, as it is often taken for granted. This conception is tied with Aristotle's perspective, which notes that Doxa consists of *those opinions [...] which are accepted by all or the majority, or the wise, or the most notable and renowned* (*Aristotle*, in Barnes [21], p. 167). This is in line with Turbanti [308], stating:

   > Human beings are communicators by nature. From birth, we are actively and passively immersed in communicative flows that shape and determine crucial aspects of our daily lives. The society in which we live has long been called a 'communication society': a society in which communication not only plays a crucial role, but also essentially characterises what it means for an individual to be part of it. We are members of our society to the extent that we have communicative relationships with each other. These relationships define our social statuses, shape culture and values, and determine economic processes and production systems" [p.7].

   This conception falls within Grice's principles of cooperation, as it suggests that successful communication is based on mutual cooperation between participants who share a common understanding, which is regarded as *Common Ground*, defined by Clark and Brennan [60] as *self-evident information that is universally acknowledged among humans* . Therefore, both Vico and Grice converge in emphasizing the importance of shared assumptions, understanding and especially cooperation from both sides in communication.

This integrated view of CS allows us to explore these features individually, starting with the distinctions between CSK and CSR that were proposed in the Scottish/English Enlightenment and have since been adopted by AI research. Although this work primarily focuses on CSK, it is important to highlight those aspects to better understand the theoretical approach presented in Chapter 4.

3.1.1 Commonsense Knowledge

Commonsense Knowledge (CSK) is a stock of knowledge which is self-evident, requires no logical or other demonstration and is widely shared and accessible. Therefore, it is commonly assumed that one's own understanding of concepts is generally shared by others, especially among individuals deemed to possess rationality [140]. Since it denotes inherently evident truths, it does not require explicit justification, leading to its implicit use in communication, both written and oral [125]. It only surfaces explicitly during ambiguous situations or when the speaker necessitates clarification [221]. Zang et al. [330][p.690] identified the most representative characteristics that provide a complete description of this type of knowledge, such as:

1. **Share**: A group of people own and share the CSK
2. **Fundamentality**: People have a good understanding of the CSK and tend to take it for granted.
3. **Implicitness**: Most of the time, people tend not to mention the CSK explicitly, as it is knowledge that is assumed to be shared among interlocutors.
4. **Large-Scale**: The CSK contains massive, large-scale information;
5. **Open Domain**: The CSK is broad in nature and covers all aspects of everyday life rather than a specific domain.
6. **Default**: The CSK includes predefined assumptions about typical cases of everyday life; therefore, most of them may not always be correct.

This knowledge is gained through direct experience with the world, as human learning lies in the ability to act on and interact with the surrounding environment [120]. This is how humans learn to encode the characteristics of objects through manipulation (*action*) and to understand the surrounding environment by mapping the regions (*interact with*) where objects are situated inside or outside the manual reach [104]. In addition, learning also takes place through interaction with other people, which is referred to as *social learning* [303], a broad term encompassing a wide range of forms of learning, including motor, verbal and knowledge learning, through various forms of social context including observations of others, imitation, and interactive learning. Therefore, CSK embodies the fundamental understanding

of the world shared among individuals, encoding information people use everyday encompassing many aspects of life including [51]: (i) **Information about events that occur over time.** Temporal knowledge pertains typically times, order, frequency, etc, and natural language rarely communicates explicit temporal information. Instead, it is vague and relies on the commonsense knowledge of the listener [272]. (ii) **The consequences of one's own and others' actions.** People are capable of making inferences about other people's mental states, which can be influenced by a set of social norms of behaviour [118, 272]. Lastly, (iii) **people have information about physical properties and affordances of everyday objects** [135, 201], along with their perceptions, properties and their interrelationships [272].

3.1.2 Commonsense Reasoning

As we have seen so far, CSK is part of human experience everywhere and anywhere, no competence beyond birth and primary socialisation is required to process it [25]. Therefore, CS also refers to everyday reasoning, helping humans navigate everyday situations seamlessly [10]. In this respect, Hammond [132] defines it as the process of making a judgement in the balance between intuitive decision-making and formal, analytical reasoning. In his work, he highlights that judgement involves trade-offs between the adaptability of intuition and the rigour of formal analysis and that finding this balance is crucial in decision-making. Intuition provides robustness with a large variance (being on target with a wider dispersion), while analysis provides accuracy but can lead to significant errors (off target with a small and biased variance). In short, intuition may be valid but unreliable, while formal analysis tends to be reliable but potentially invalid. In this regard, CS addresses precisely the tension between robustness and accuracy, or reliability and validity, in decision-making as it is *quasi-rational thinking*, which allows for quick decisions, especially in the presence of constraints such as time, resources and uncertainty [25]. In this regard, Commonsense Reasoning (CSR) refers to the cognitive process that enables humans to connect implicit knowledge to draw new conclusions [211, 68]. In other words, is the human-like ability to make presumptions about the types and essence of ordinary situations that humans encounter daily. These assumptions include judgments about the nature of physical objects, taxonomic properties, and people's intentions [68]. A device exhibiting CSR might be capable of drawing conclusions similar to humans' folk psychology, i.e. their innate ability to reason about people's behaviour and intentions, as well as their

naive physics, i.e. their natural understanding of the physical world. In this field, facts are commonly represented in a declarative language such as first-order logic, which is a collection of formal systems used in mathematics, philosophy, linguistics, and computer science, using quantified variables over non-logical objects, and allows the use of sentences that contain variables [3]. Indeed, CSR is the automatic deduction of necessary consequences to solve a problem and involves various reasoning methods, such as analogical, statistical, logical, and heuristic reasoning [280].

3.1.3 The Interplay Between Commonsense Knowledge and Reasoning

McCarthy argued that a computer's capacity for CS could best be achieved by using a declarative representation of CSK and some means for deducing its consequences. That is, to get a computer to exhibit CS, one must first provide it with a substantial amount of basic facts about the world. In this respect, Knowledge Representation (KR, which, in this context, refers to CSK representation) and the automation of CSR have been intrinsically linked since then [216]. Whatever the exact nature of the connections between these two communities, the past several decades have seen a great deal of activity in both KR and CSR. Particularly since the late 1980s and early 1990s, there has been growing emphasis on synthesizing work across different groups of researchers studying a given topic [216]. For instance, numerous results have compared the expressivity of various languages and theories of action [269], or compared different description logics [14], with the goal of developing standards and criteria for evaluating such research. However, those in the CSR community are primarily focused on object-level formalization, rather than meta-domain research [4]. In other words, serious engagement in object-level formalization is a defining characteristic of membership in the CSR community, beyond simply being part of the KR community [216].

[3] For instance, in first-order logic, rather than using simple propositions like "all men are mortal," one can have more complex expressions in the form "for all x, if x is a man, then x is mortal." Here, "for all x" is a quantifier, "x" is a variable, and "... is a man" and "... is mortal" are predicates. This distinguishes first-order logic from propositional logic, which does not use quantifiers or relations. In this sense, propositional logic serves as the foundation for first-order logic. Wikipedia: https://en.wikipedia.org/wiki/First-order_logic [last visited on 21/02/2025].

[4] In this respect, one of the major exponents of CSR among other scholars is Leora Morgenstern (cited in this work), who manages the Common Sense Problem Page, initially created by Rob Miller. As stated on their page, the main goal of this community is to codify CSK using formal logic. For further information, it is possible to consult their main page at following link: http://www-formal.stanford.edu/leora/commonsense/ [last visited on: 6/02/2025]

Both fields of KR and CSR involve the development of formal languages and structures for representing knowledge, as well as methods for reasoning with that knowledge (more theoretical details about KR will be provided in Chapter 4). Nevertheless, though closely related, these two fields are not identical. Rather, they are linked through their shared communities and a symbiosis of problems and results, but they are distinguished by their differing focal interests [216]. The KR community has other sources for its research problems, most notably the study of large amounts of information in knowledge base systems, and querying those knowledge bases about simple consequences of the stored information. This has led to work on frame-based systems, semantic networks, description logic, and various proof methods already mentioned in Chapter 1. Additionally, the question of how to develop expert systems has spurred the development of rule-based systems and accompanying proof strategies. These research areas are generally not considered to be part of CSR. Importantly, CSR and KR are interconnected, with CSR feeding into KR and KR feeding into CSR. The CSR community employs results developed in the KR community, such as methods for translating between temporal languages [216]. The KR and CSR communities have different research focuses. CSR researchers tend to be more concerned with the intricate details of axiomatizing specific domains, while KR researchers tend to focus more on meta-domain research, such as demonstrating equivalence between various formalisms and investigating complexity issues. This is not to say that CSR researchers are uninterested in meta-domain research. Rather, those engaged in rigorous, object-level formalization are almost certainly members of the CSR community, and not solely part of the KR community [216]. However, there is considerable overlap between KR and CSR. Researchers working on theories of causation, for example, or on the interaction between knowledge and ability, can equally describe their work as KR or CSR. Consequently, there is a substantial intersection between the KR and CSR research communities. Conversely, CSR and KR mutually inform one another. CSR can be viewed as a source of—and even a breeding ground for—research problems that drive the KR community. The development of theories of action, causation, planning, knowledge, and default reasoning began as attempts to formalize commonsense reasoning problems before these became established subareas of KR [216]. This overlap will be taken into account to support the theoretical framework introduced in Chapter 4 and the implementation of conversational agents in Chapter 6.

3.2 Knowledge in Cognitive Linguistics

In linguistic literature, it is possible to find various conceptualizations of knowledge. These include Encyclopedic Knowledge, i.e. knowledge of the world, and Common Ground, i.e. mutual knowledge, beliefs and assumptions that people rely on to communicate efficiently. First, these two concepts will be described individually. Then, the differences and similarities between these concepts and CSK will be examined.

3.2.1 Encyclopedic Knowledge

The term Encyclopedic Knowledge (EK) is used to denote knowledge of the world, referring to a broad set of knowledge that a speaker associates with the concept expressed by a word, a knowledge derived from their experience of the world (also often referred to as *world knowledge*) [192]. The encyclopedic view represents a model of the conceptual knowledge system underlying linguistic meaning. Therefore, the meaning of a single word cannot be understood without access to all the essential knowledge - linguistic and extralinguistic - related to it [152]. Cognitive semanticists conceptualise EK as a structured network system, playing a crucial role in the way human beings make sense of communication [192, 90]. However, not all aspects of the knowledge accessible through a single term hold equal meaning. For instance, the EK associated with the word *bread* encompasses the operations one can perform on that object (e.g the bread can be cut), its transient characteristics (e.g. bread can be fresh, stale), and so on, regardless of one's personal attitudes towards the object in question [152, 90]. Encyclopedic meaning emerges within the context of use, with the 'selection' of such meaning influenced by contextual factors [90]. In contrast to the dictionary perspective, which separates core and non-core meanings, the encyclopedic view holds that semantics encompasses EK, and meaning is fundamentally 'guided' by context. Furthermore, the meaning of a word is 'constructed' dynamically as a result of contextual information. From this standpoint, fully-specified pre-assembled word meanings do not exist, but are instead selected and formed from EK [90]. Indeed, the encyclopedic approach conceptualizes lexical items as gateways to EK [177]. According to this view, words are not self-contained receptacles that present fixed bundles of information. Instead, they selectively grant access to specific portions of the extensive network of EK. While the core meaning of a word tends to remain relatively constant, the EK that each word affords access to is dynamic [90].

Taking into account the lexical concept of *car*, the understanding of the concept of it develops over time as people interact with cars and accumulate more knowledge about them [22, 90].

3.2.2 Common Ground

As we have seen in Chapter 2, Grice [125] emphasizes the critical role of cooperation in enabling successful communication by anticipating the concept of *Common Ground*. Indeed, for interlocutors to understand each other effectively, they must share a grounded knowledge, which becomes an integral part of communicative processes [124][p.65]. In Stalnaker's words [287]:

"When speakers speak, they presuppose things and what they presuppose guides both what they choose to say and how they intend to interpret what they say. To presuppose something is to take it for granted as background information, as common ground between the participants in the conversation" [p. 701].

As pointed out by Clark [59], Common Ground (CG) refers to the mutual knowledge, beliefs, and assumptions that people rely on to communicate in an efficient way. The concept of Common Ground is crucial for successful communication, as it allows interlocutors to build a shared understanding and frame of reference. In this respect, it is possible to define four main types of CG [59]:

1. **Personal Common Ground (PCG)**: it is established through continuous communicative interactions with an interlocutor and is considered as a register of shared experiences with that specific speaker;

2. **Local Common Ground**: Part of the PCG is contained in the Local Common Ground, which is based on information from a single interaction with known or completely unknown interlocutors. An example is asking for the timetable of trains and shops;

3. **Communal Common Ground (CCG)** comprises the knowledge shared within a community. It is the amount of information shared with people who share general knowledge, knowledge related to social background, education (schools attended, levels of education attained), religion, nationality and language(s);

4. **Specialised Common Ground (SCG)**. Within the GCC, one can find a more specific community, defined as Specialised Common Ground (SCG). This is defined among people with shared expertise in a particular field, such as colleagues in a specific field (e.g. medicine) and is characterised by a specific vocabulary.

Therefore, CG is a cooperatively constructed mental abstraction, which is assumed by the interlocutors in the sense that no one will know for sure that it exists [164]. In this regard, all collective actions rely on CG and its continuous accumulation and establishment. Updating CG requires a process known as *grounding*, essential to effective communication [61]. To pursue it, two conditions must be met: (i) the contributor specifies their message's content, and (ii) the partner acknowledges and records that content. The grounding process ensures that both parties mutually believe they have understood the message, allowing CG to accumulate and form a contribution [61]. In conversation, a contribution is pursued when speaker *A* presents an action for speaker *B* to consider, and *B* accepts that action as having been understood. If these two steps have been performed correctly, *A* and *B* will consider that they have come to the mutual belief that *B* has understood what *A* meant by his action [61].

3.2.3 Differences with Commonsense Knowledge

Although EK and CSK share knowledge gained through direct experience of the world, EK deals with semi-structured, explicit, factual and domain-specific information, encompassing information usually found in news, articles, debates, lectures, etc., but also principles and definitions that can be found in encyclopedias and structured resources [49]. In this regard, from a linguistic point of view, is highlighted the distinction between EK and dictionary knowledge, i.e. division between ontology and lexicon. Dictionary knowledge is supposed to cover the idiosyncrasies of particular words, whereas EK covers everything about the underlying concepts [164]. Thus, EK is dealing with information regarding attributes (e.g. age, duration) of entities (e.g. person, event) and relationships (e.g. educated to, followed by) between entities [328]. However, when people communicate with each other, they are also relying on similar background knowledge. i.e., they are dealing with everyday events and their effects (e.g., cleaning the floor if food stood on it), facts about beliefs and desires (e.g., studying hard to win a scholarship) and properties of objects (e.g., an egg is composed by yolk, white and shell) [328, 49]. This taken-for-granted information is

what is defined as CSK. Therefore, unlike EK, which provides precision and details for specific domains, CSK applies to general contexts and is usually shared by most people and assumed implicitly in communication [328]. This means that it does not involve the grounding process already seen for CG to be established. In this respect, despite both CG and CSK involve shared understandings and assumptions about the world, these are essentially distinct concepts. As highlighted in Mennella [204], three features delineated in Zang [330], respectively (i) sharedness, (ii) fundamentality, and (iii) implicitness appear interesting as they recall some aspects of the CCG. (i) Sharedness refers to the mutual knowledge or assumptions that participants share in communication, involving cultural knowledge (e.g., a common language) [195]. In any communicative interaction, people rely on this shared information to make the conversation efficient and coherent. It is important to emphasise that sharing in this context is not universal, but group-specific. as highlighted in Whiting et al. [324], different communities may have different common knowledge, which has an impact on shared beliefs among group members that they believe to be universal. Therefore, while CCG implies a specific connection between an individual and other members of a shared community, emphasising the interaction between the interlocutors, CSK is concerned with an individual's interaction with the world at large. As pointed out in the work by Whiting & Watts [324], it is also related to what people have acquired through direct experience and how this knowledge is believed to be already shared with other individuals. (ii) Fundamentality refers to basic, deeply rooted knowledge or assumptions. As mentioned earlier, CSK refers to beliefs, values and norms shared within the community, being cultural assumptions or social norms fundamental to understanding within the community. Although CSK information is generally shared within a culture, it may vary between different communities or societies, as some knowledge may not be universal (e.g. social norms). Therefore, CSK information is much more general and goes beyond the cultural context of the individual. The force of gravity, as conceived in Hayes [135], is an intuitive physical concept widely shared by human beings and unrelated to formal education. This intuitive understanding is based on a set of fundamental principles derived from everyday experience. Since this information is deeply embedded in the knowledge of speakers, (iii) implicitness refers to the unstated (e.g. not verbalised) and assumed information that participants rely on during communication. This information is so fundamental that it is rarely questioned or explicitly discussed and forms the basis of much of the communication

[272]. Another difference between CCG and CSK lies in the process of defining this information. Indeed, while CCG involves active agreement between speakers, establishing shared beliefs and defining a common language for group identities and boundaries [195], on the other hand, CSK does not require any explicit agreement for the aforementioned motivations.

3.3 Conclusions

Common Sense (CS) is defined as a natural ability to make good judgments and to behave in a practical and reasonable manner. The term is widely used across diverse contexts, such as philosophical discourse, history, sociology, psychology, and the assessment of AI. From it, we can extract three key concepts that delineate the notion of CS: (i) CS solves the "binding problem", i.e., integration of multi-modal sensory experiences; (ii) CS is a stock of universally recognisable knowledge to reason from and is the quasi-rational judgement call: neither pure intuition nor formally rational (iii) CS is the communal sensitivity that binds us into joint attention and intentionality. This integrated view of CS allows us to explore these features individually, starting with the distinctions between Commonsense Knowledge and Commonsense Reasoning that were proposed in the Scottish/English Enlightenment and have since been adopted by AI research, Commonsense Knowledge (CSK) represents a stock of knowledge that is self-evident, requiring no logical or other formal demonstration, and is widely shared and accessible. Commonsense Reasoning (CSR) pertains to everyday reasoning, helping humans navigate everyday life situations seamlessly. In the field of AI, Knowledge Representation and the automation of Commonsense Reasoning have been intrinsically linked since the advent of McCarthy's work in 1969. The goal has been to develop standards and criteria for evaluating the capacity of machines to *understand* the world around them and solve problems. CSK is related to, yet distinguishable from, another body of knowledge known as Encyclopedic Knowledge, which denotes a broad set of world knowledge that a speaker associates with a concept, derived from their lived experience. While EK and CSK both involve knowledge gained through direct experience of the world, EK deals with semi-structured, explicit, factual, and domain-specific information. In contrast, CSK encompasses everyday events and their effects, as well as facts about beliefs, desires, and object properties. To communicate successfully, people employ another stock

of knowledge known as Common Ground (CG), which refers to the mutual knowledge, beliefs, and assumptions that facilitate efficient communication. Specifically, Communal Common Ground (CCG) encompasses the knowledge shared within a community, such as information related to social background, education, religion, nationality, and language. Although CSK is generally shared within a culture, it may vary among different communities or societies, as some knowledge may not be universal. Additionally, while CCG involves active agreement between speakers, establishing shared beliefs and defining a common language for group identities and boundaries, CSK does not require any explicit agreement. The development of conversational agents is fundamental in this field, as one of the main objectives is to represent information about the world in a form that enables the system to resolve complex tasks, which depend on the application domain. In this regard, we have observed that CSK is closely intertwined with CSR. While CSK refers to facts acquired through experience and/or learning, and its representation implies formalizing this information to make it understandable to machines, CSR involves deducing the necessary knowledge from what is explicitly available. Given the complexity of this knowledge, CSK will be conceived as a dynamic process that emerges from communication, leading to the development of a theoretical model that seeks to analyse the underlying processes directly from linguistic data. This framework allows for a more structured approach to management, facilitating clearer understanding and more effective data analysis.

CHAPTER

FOUR

EXPLORING THE STRUCTURE OF COMMONSENSE KNOWLEDGE

This chapter outlines the concept of conceiving CSK as a dynamic process that emerges and evolves through communication, rather than viewing it as a static repository of information. Building on Saba's idea [265, 264] that the structure of this knowledge should be discovered rather than imposed, three levels of analysis are proposed to investigate the structure of this process, drawing on insights from cognitive linguistics.

4.1 Knowledge as a Process

As highlighted in Chapter 1, developing a conversational intelligence system with CS capabilities has been a primary goal in the field of NLP [123, 289, 143, 332]. To pursue successful communication, there is a set of normative principles that manage the dialogue according to logical and relevant criteria while respecting the principle of cooperation between the speakers [125]. The maxim of quality states that the contribution to the conversation should be as informative as required. Therefore, a speaker is not expected to provide an excess or deficiency of information; rather, they will offer only the necessary information. Consequently, people usually assume a division of the knowledge they share. In this regard, although some information is introduced explicitly into the discourse, other information, such as that related to CSK, is instead assumed and not explicitly discussed, agreed upon or questioned [5]. Knowing what can be taken for granted and what needs to be made explicit, in other words, means demonstrating communicative competence [145], which is still a

challenge for conversational agents. Several efforts have been made to represent this knowledge and organize it for computational purposes. However, many challenges have been encountered, including problems of scalability, creating complete and accurate datasets, integrating different sources, and managing inconsistencies and gaps among them. In particular, there was a dispute over the representation of knowledge supported by Rodney Brooks [43], which argued that *it is better to use the world as its own model*, rejecting the possibility of being able to create a database that could contain information that captures all aspects of human knowledge. Brooks is one of the supporters of the embodied approach, which emphasises the importance of perception, action and physical interaction with the environment in learning and cognition. On the other hand, the advent of LLMs has led to a marked improvement, allowing better actions and sequences to be generated to perform specific tasks without relying on external knowledge or similar constraints, suggesting the ability to achieve such results even under conditions of increased uncertainty and randomness. As pointed out in Bauer [25][p.26], since the machine produces plausible arguments based on stochastic models of language, people confuse the ability to produce language with comprehension, becoming 'stochastic parrots' themselves. In this regard, the famous Chinese room thought experiment, proposed by Searle in the 1980s [277], rails against the computationalism approach, which posits that the human mind can be reduced to a set of symbolic operations. In contrast, Searle's work supports a more cognitive perspective on the nature of the mind.

The continuing need for progress in equipping conversational systems with robust and adaptive CS capabilities and the obstacles posed by both generative AI and KBs form the basis of this work. In this respect, Saba [265, 264] specifically argues that *a CSK structure should be discovered rather than created*, emphasizing that it is not enough to simply establish some principles for the ontological design of this knowledge; rather, it is essential to develop a strategy for the systematic and objective design of a CSK ontology. From this line of thought, this work proposes to frame this knowledge as a *process* rather than a static representation of facts. This thought is highlighted also in Evans [90], which states that **meaning is a process**, not a discrete 'thing' that can be 'packaged' by language, as language itself does not encode meaning. Rather, words serve as *cues* that guide the construction of meaning. This concept aligns with the broader tradition of pragmatic reasoning in language,

originally introduced by Grice [125] and already discussed in Chapter 2, posing that speakers are cooperative and choose their utterances to convey specific meanings.

Given the large amount of implicit information widely assumed by humans and the impossibility of representing it fully and exhaustively in a single resource, in this work, CSK is not necessarily confined to explicit representations of knowledge relations but can emerge through communicative interaction and be reconstructed through entities that show meaningful correlations within high probability.

As already mentioned in Chapter 3, knowledge is strongly linked to reasoning. Knowledge consists in knowing something obtained through experience and/or learning, while its representation involves formalizing the acquired information to make it understandable to machines. On the other hand, reasoning entails deducing necessary knowledge from what is explicit in the available knowledge [1]. It is referred to *automated reasoning* when using AI methods such as knowledge representation, inference, heuristic search, and machine learning. In the case of conversational agents, in selecting the next action to be useful and appropriate, the agent must reason about the available actions, and their expected and unexpected effects, estimate their feasibility, and consider the objectives. In this regard, the foundational idea of assumption-based reasoning stems from Bernard Bolzano's work [262] on the variation logic. From this perspective, every proposition is either true or false and retains that status indefinitely. However, in some cases, the same proposition can paradoxically be both true and false. According to Bolzano, the reason is that in the original proposition, some components may not have been explicitly stated in the corresponding linguistic expression, as some of them have been altered in some way [2]. In other words, the mutable components of the proposition are assumptions that can be true or false, thereby influencing the truth of the entire proposition [163]. It is in line with what concerns the human capacity to consider which assumptions underlie a given conclusion and to modify that conclusion when those assumptions change. In the computational domain, the management of these dependencies between assumptions and conclusions

[1] This theoretical aspect on reasoning is gathered from the Master's Thesis in Computer Science carried out by Danilo Esposito, supervised by Prof. Antonio Origlia. This aspect is crucial as it supports the implementation of conversational agents discussed in Chapter 6.

[2] Bolzano's Logic. Stanford Encyclopedia of Philosophy: https://plato.stanford.edu/entries/bolzano-logic/EarlWorkLogiMeth [last visited on: 22/03/2025]

is facilitated by a Truth Maintenance System (TMS), which is essentially a knowledge representation system and a solver [3]. The solver's knowledge base allows for the addition of new information, which can invalidate previously drawn conclusions but also justify new ones. The nature of the problem necessitates that the TMS be able to track the relationships between conclusions and the arguments supporting them. If some of these arguments become invalid due to changes in the knowledge base, the corresponding conclusion should be invalidated as well [163]. The assumption-based reasoning concept allows for the discarding of arguments that are no longer valid, enabling the computation to continue based on the remaining valid arguments. This differs from the use of mathematical logic, where even false arguments can be used to reach tautological conclusions [28]. Kean et al. [163] elaborates on the formal, primarily logical theory behind systems that reason according to hypotheses, which encompasses concepts such as direct consequence, justification, set of conflicts, extension, agreement, and irrefutability. Assumption-based reasoning has practical applications in both standard and non-monotonic logic programming. In the latter case, it employs AnsProlog [4] , a subset of Prolog [5] designed for Answer Set Programming, which is particularly useful for solving planning and knowledge representation problems. In this context, Kean et al. [163] employ this type of reasoning to demonstrate the feasibility of modelling a portion of human knowledge. When no information is available about a goal g, i.e. no rule is defined for g, it is possible to consider two distinct scenarios: (i) g is true and (ii) g is false. Each computational branch is then extended according to the other rules of the respective scenario. In this regard, it is important to consider the reasoning under uncertainty. There are disagreements regarding its definition. As reported in Walker [316], within the regulatory and management sciences, there is neither commonly shared terminology nor

[3]For more details on this aspect, see Doyle, 1979 [78].

[4]" Answer Set Programming (ASP) is a declarative programming paradigm with a semantics known as the answer set semantics [19]. It is declarative in that the programmer specifies what needs to be achieved, rather than how it should it be achieved. It therefore lends itself naturally to applications in the domain of artificial intelligence, such as plan generation and reasoning in agents. ASP programs, which are written in the language of AnsProlog*, are composed of a set of facts together with a set of rules from which other facts can be derived. A set of consistent facts that can be derived from a program using the rules is known as an answer set of the program. The possible answer sets for an AnsProlog* input program are computed with a program called a solver ". Sureshkumar, 2006 [296] [p.1].

[5]"There are only three basic constructs in Prolog: facts, rules, and queries. A collection of facts and rules is called a knowledge base (or a database) and Prolog programming is all about writing knowledge bases. That is, Prolog programs simply are knowledge bases, collections of facts and rules which describe some collection of relationships that we find interesting. So how do we use a Prolog program? By posing queries. That is, by asking questions about the information stored in the knowledge base." Blackburn et al., 2006 [34].

full agreement on a typology of uncertainties:

> "[...] We adopt a general definition of uncertainty as being any deviation from the unachievable ideal of completely deterministic knowledge of the relevant system."[p.1]

Uncertainty is colloquially presented in statements of likelihood. For example, it may be stated that *it is very likely that the cake will rise properly* or *it is very likely that the pasta will be overcooked if left boiling for too long* [2]. In the computational field, there are two ways to address uncertainty: (i) it can be managed implicitly, ignoring it as much as possible or creating robust procedures for handling it or (ii) it can be managed through explicit methods, building a model that describes the uncertainty. The construction of such a model involves defining various aspects, including language, semantics, query design, and response modes, among others. This leads to the concept of reasoning under uncertainty or probabilistic reasoning. In this respect, there is an implicit assumption in all that follows that uncertainty can be measured by probability. Indeed, there is a very good argument that probability is the best measure of uncertainty, as reported in the work conducted by Lindley [189]. In this respect, when dealing with probability, we refer to the measure of the possibility associated with the occurrence of a random event, i.e., an event that occurs randomly which is not certain [153]. Probability calculations assign numerical values, often decimals between 0 and 1, to each event in the sample space. In this context, 0 indicates an impossible event, while 1 indicates a certain event. The sum of the probabilities of all events in the sample space, whether simple, complex, or joint, is always equal to 1. The set of probabilities assigned to each event in a random process is known as the probability distribution. A distribution describing all events as equally probable is defined as uniform. Moreover, the calculation of probability is closely linked to the concept of entropy, as defined in information theory. Entropy measures how informative a set of events is or how predictable it is that a specific event will occur in that set. The fundamental concepts in probability theory are: (i) *prior*, i.e., the probability of an event before further information or observations. It is usually denoted as P, where A represents the event; (ii) *conditional*, i.e., the probability of an event B given that another event A has already occurred. Denoted as P 'the probability of B given A'; and (iii) *joint probabilities*, which pertain to the joint probability of two events A and B, representing the probability that both events occur simultaneously. The symbol is

used to represent the intersection of events A and B [153][p.78-80]. In support of this view, Verschueren et al. [312] point out that information from long-term memory is of two types: (i) frequency or likelihood estimations (e.g. probability) information; and (ii) counterexamples information, which pertains to person's general knowledge about the world [47]. This is in line with Bybee [46], which states that language arises from a general, usage- and frequency-sensitive learning mechanism, rather than relying exclusively on innate, domain-specific structures, as supported by generativists. This view fits well with Bayesian cognitive modelling, which has been the focus of many recent attempts to understand the interaction between structured representations and graded or statistical information [301, 121]. Bayesian networks are probabilistic graphical models that represent causal relationships between a set of variables through the use of an acyclic-oriented graph [151, 288]. These tools were first introduced by Judea Pearl, an American computer scientist and philosopher, with the publication of the book '*Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*' in 1988 [239], representing a turning point in their introduction and development. In this work, Pearl introduced key concepts that shaped the field, including the graphical representation of Bayesian networks, the use of conditional probabilities to model relationships between variables and the concept of probabilistic inference. These models are used to model and represent probabilistic relationships between variables in a system and are widely used in various fields to address problems of uncertainty and probabilistic reasoning. Indeed, these models have been an important tool for understanding the non-linguistic varieties of rational action, integrating belief understanding with action planning [15]. Specifically, this aspect will be essential for the application described in Chapter 6.

In this respect, graph databases offer a means to represent and navigate complex networks of information, reflecting the dynamic nature of human knowledge representation. Indeed, graph databases have gained popularity in recent years due to their ability to represent complex data in an interpretable and flexible format. They prove particularly useful in linking different resources, enabling the integrated analysis of information from multiple sources [228]. Therefore, data representation in the form of a graph, traced paths and the likelihood of finding specific types of relationships enable information extraction, reflecting the human approach to knowledge in a more dynamic way. In this study, Neo4j, a complete and self-contained graph database management system, has been employed as it is specifically designed to store, manage,

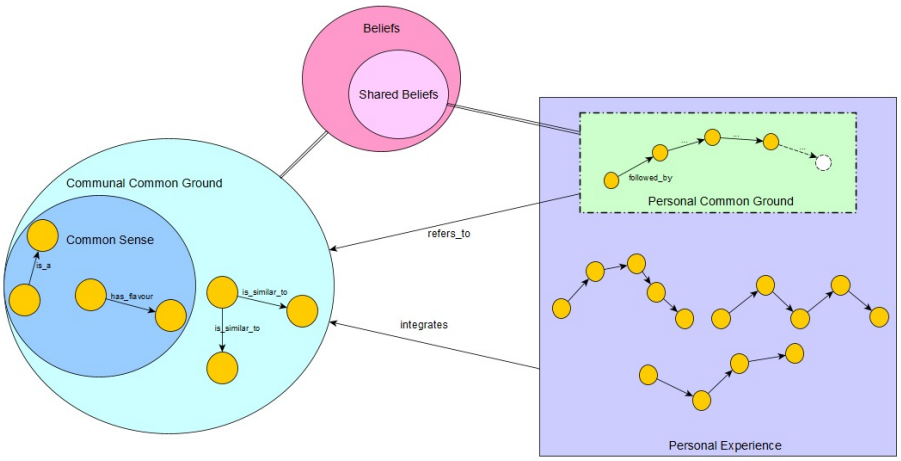

FIGURE 4.1: Knowledge graph representation divided into common ground, personal common ground, personal experience and beliefs. Although they are separate concepts, they are always interconnected.

and query data in graph form [323]. It allows one to define nodes and relationships between them, employing Cypher queries [105] to explore the graph structure for complex information retrieval [323]. In this work, CSK is not necessarily represented by relationships established in a knowledge base but can be derived and reconstructed using probabilistic approaches, becoming any kind of graphical representation that a speaker believes exists in the knowledge possessed by the interlocutor with a high probability, as shown in Figure 4.1. Based on the observations reported in Chapter 2, CS appears to be an integral part of CCG [6].

4.2 Three Analysis Level

Since ordinary language is the best-known theory we have of everyday cognition [264], the development of a model that seeks to extract the underlying processes directly from linguistic data seems to be more human-like [7]. In this regard, information are divided into three macro categories, allowing for a more structured approach to managing information, thereby facilitating a clearer understanding and more effective analysis of the data.

[6]These observations are the result of a collaboration with Dr. Maria di Maro, who is specialized in Common Ground inconsistencies management in dialogue systems and is credited with creating the graph presented in this work. More about this topic can be found in her work: Di Maro, 2021 [73].

[7]Some aspects of the proposed theoretical framework is described in the work carried out by Mennella et al., [205] and Mennella [203].

1. **Foreground Knowledge**. Foreground knowledge is defined as information explicitly expressed in both oral communication and written texts. The definition here relates to Fillmore's semantic frames [101], where the (explicit) linguistic elements evoke a semantic frame;

2. **Background Knowledge**. Background knowledge pertains to basic fundamental information about entities often left omitted in communication. The definition is taken from the work pursued by Minsky [210] and Cambria [48, 50], who refer to this information as belonging to CSK. In this work, however, this knowledge refers to cognitive frames that give access to a vast amount of information structured by it. Cognitive frames are essential to language, as speakers depend on them to understand words and are necessary background knowledge for interpreting semantic frames;

3. **Presupposed Knowledge**. Presupposed Knowledge is categorised as the implicit information automatically inferred by speakers. This information is in line with the definition given for the communicative frames, which pertains the process of selecting and emphasising certain aspects of a perceived reality within a communicating text [85]. Moreover, they are defined as *presupposed* as in the interpretation of the sentence, the principle of pragmatic presupposition complements the semantic representation [185].

These knowledge are interrelated, as the former facilitates the accurate interpretation of the latter. This is particularly evident when considering foreground and background knowledge, which aligns with Fillmore's explanation of the relationship between cognitive and semantic frames, defining the cognitive frames as *those background understandings needed for making sense of things that happen around us*, which are activated by a wide range of information. On the other hand, semantic frames are *those that are specifically coded in—or "evoked by"—lexical units or other features of linguistic form* and are evoked only by units of language [7] [p. 158]. While background knowledge is considered essential for interpreting utterances, the verbal discourse itself and the context of the utterance are equally crucial. Therefore, background knowledge should be considered alongside the semantic representation of a sentence spoken in a particular context [233]. As highlighted in Sullivan [295], semantic frames and cognitive frames contribute to communicative frames, which serve to promote a particular problem definition, such as a causal interpretation for

the item described [85].

For this work investigation, the cooking domain is taken into account, guided by two main factors:

1. Culinary practices are presumed to be highly familiar due to their everyday nature as most people routinely prepare meals. Therefore, it is assumed that people are aware of both the basic tools for preparing, cooking, and storing food as well as the physical state of the ingredients and their change-of-state process. For instance, an egg consists of an inner liquid (the yolk and albumen) and a shell that contains it. Depending on the cooking method, the inner liquid changes its initial state, e.g. from liquid to solid if it is boiled.

2. The domain exhibits strong action co-occurrences, as individual actions are linked. For instance, the action of *beating eggs* inherently implies that the eggs have been previously cracked. In this context, actions of this kind are mostly omitted in cooking recipes as long as dialogues, as they are considered fundamental and mandatory steps mainly because of the systemic nature of their occurrence. This omission reflects a deep and shared understanding within the domain, where the sequence of actions is so predictable that details are often assumed rather than explicitly stated. This simplifies communication, focusing attention on the most relevant or variable aspects of the process.

At the state of the art, many works have focused precisely on this issue [8]. In particular, Morgenstern's work [215] discusses some fundamental theories of containment, falling and pouring, integrated into Shanahan's circumscribed calculation of events, and shows how these can serve as the basis for an axiomatization that partially characterizes the breaking of eggs. Here, however, I propose to describe this procedure from a purely linguistic perspective: through instructions for *whisking the eggs* may not explicitly mention it, we inherently infer essential *presupposed knowledge*, including prior actions like *egg-breaking* and the use of a tool (e.g., a fork) for the beating process, as long as the *background knowledge* about the nature of eggs themselves (e.g., eggs are liquid and can be beaten).

[8]Davis, E., 'The egg cracking problem', Commonsense Problem Page: http://www-formal.stanford.edu/leora/commonsense/eggcracking.html [last visited on: 06/03/2025]

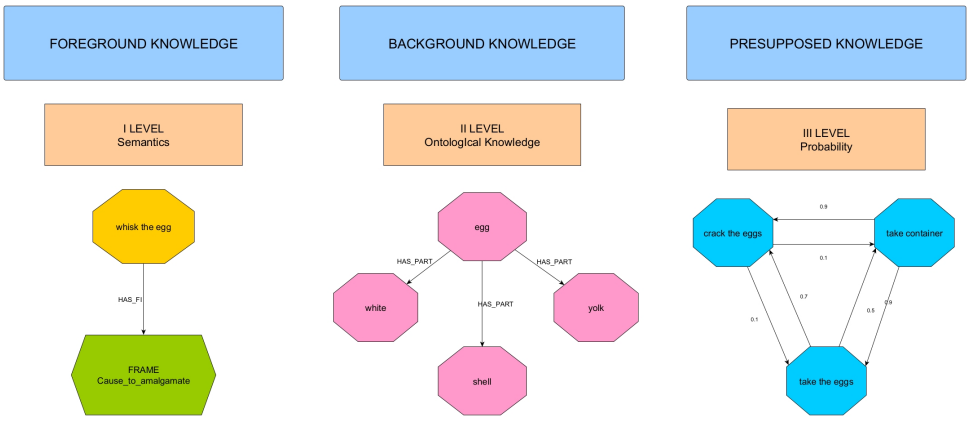

FIGURE 4.2: Analysis model with the example of the instruction *whisk the eggs*. At the first level, the action *whisk the eggs* invokes the *cause_to_amalgamate* frame. At the second level, the model includes ontological knowledge of entities (e.g., *egg*) and their subparts (e.g., *shell*, *yolk*). At the third level, the action of *whisking* implies a series of action chains (e.g., *take container*, *crack eggs*), determined by the probability of their occurrences represented as relationship properties.

To uncover the processes underlying the foreground information, a three-level analysis is proposed and summarised in Figure 4.2:

- **I Level**. The focus is on the action and the entity involved in a foreground event. This level refers to a semantic analysis applied to each sentence within the CookDial dialogical corpus [157], employing an annotation scheme based on FrameNet. In FrameNet, each concept (*frame*) is schematised along with its definition, its examples, and its *frame elements*, which represents the semantic roles required by the *lexical unit* (LU) evoking the frame. An example is represented by the sentence *whisk the eggs*, where the action of *whisking* evokes the frame *cause_to_amalgamate* described in FrameNet.

- **II Level**. This level relates to the comprehensive knowledge about entities and actions. This knowledge is represented by sources integrated into the graph database. The action *whisk the eggs* assumes that the knowledge of the entity *egg* is already available for the hearer, regardless of whether it has been explicitly described in the dialogue or not. This assumption is based on the fact that the knowledge of the object is part of the shared understanding of the world. This information can be retrieved by querying the database when necessary (e.g., I need to know the state of an ingredient to perform actions).

- **III Level**. The focus is on the presupposed action and entities that enable the

frame identified in the I level to take place. This level pertains to a probabilistic analysis, ultimately aiming to predict the core action that defines the frame itself. For instance, the action of *cracking the eggs*, which does not appear explicitly in the dialogue, is implied in the action *whisking the eggs*, semantically marked as *cause_to_amalgamate*. By applying the probabilistic estimation on entity relationships within the database, it will be possible to extract the most likely co-occurrences of actions within a given semantic context, while avoiding the verbalisation of presupposed actions.

4.3 Cooking Domain Sources

To simplify the process of extracting implicit information for systems, a domain knowledge base is built taking into account three domain sources, respectively Recipe1M+ [198], FlavorDB [111], and Epic Kitchens [66]. Those collectively represent the knowledge base of ingredients, recipe titles with instructions, food flavours, and daily activities performed in the kitchen that are not explicitly mentioned in recipe instructions (e.g., *take eggs - crack eggs - throw eggshell into bin*). In particular, Epic Kitchens enables the retrieval of co-occurrences between actions, mapping the various steps on each entity involved. These entities are connected to the previously referenced resources, providing a comprehensive view of both entities' categories and actions. To provide insights into the categorical distribution of ingredients on which actions extracted from Epic Kitchens are performed, the frequency of actions performed on ingredients belonging to specific categories within FlavorDB will be quantified. In a dialogue, core information is typically made explicit in conversation, following a logical sequence. For example, when preparing an omelette, the action of *beating the eggs* is expected to be mentioned first, followed by putting them in the pan to cook, rather than in the reverse order. This order reflects the inherent structure of the cooking task, in which each step builds on the previous one. This sequence ensures clarity and consistency in communication, as it aligns with the natural flow of events and shared understanding of the process. Thus, linguistic analysis is conducted on the CookDial dialogical corpus [157] for both identification and distribution of semantic information along the dialogic flow. Moreover, to distinguish verbalized actions from those left implicit, a descriptive statistical analysis of the Epic Kitchens data is conducted to explore the linguistic features that characterize them.

The following sections present a description of the data taken into account for this investigation. It is important to note that, while the specific data employed here cannot be directly distributed due to license restrictions, their open-source nature allows for independent replication of the analysis, following the methodology outlined in Chapter 5.

4.3.1 RECIPE1M+

The Recipe1M+[9] [198] dataset encompasses over one million labelled recipes, offering a rich repository of culinary knowledge. Each recipe entry is structured, providing essential details including a unique identifier, a descriptive title, instructions and a comprehensive list of ingredients. Ingredient information incorporates both the quantity and the corresponding unit of measurement. Detailed instructions, segmented into steps, guide users through the preparation process. This dataset has been taken into account as a foundational ontological framework within the domain of recipe classification.

4.3.2 Epic Kitchens

The Epic-Kitchens dataset [10] [66] is designed to evaluate algorithms for recognising and understanding activities performed in a kitchen, based on video recordings made in 45 real home environments, comprising videos of people preparing food. The recordings cover everyday culinary actions and present challenges such as the heterogeneity of home environments, the variety of utensils and the variety of ingredients being used. The videos were subsequently annotated by the participants themselves, who were asked to describe the individual actions that were performed in the videos. The dataset allows for the extraction of a transition matrix that illustrates the relationships between specific nouns and the actions associated with them. This matrix serves as a valuable visualization tool, clarifying which actions can be performed on each object and providing a structured representation of these interactions. Since the corpus consists of generic actions related to cooking processes and everyday kitchen activities, it was manually labelled to distinguish between core actions (explicitly

[9]Recipe1M+: https://im2recipe.csail.mit.edu/
[10]Epic-Kitchens: https://epic-kitchens.github.io/2025

expressed) and non-core actions (left implicit), examining linguistic differences that characterize them.

4.3.3 FlavorDB

FlavorDB [11] [111] is a flavours dataset comprising structured data on the sensory characteristics of foods, including information on chemical compounds responsible for taste and aroma, and associations between ingredients and flavour profiles. It contains 25,595 flavour molecules representing a range of tastes and odours. Among the molecules listed in the database, 2,254 were identified as being contained in 936 natural entities and ingredients. The characteristics provided in the detailed molecular and taste profiles of these compounds influence their taste and smell through gustatory and olfactory sensory mechanisms. FlavorDB includes categories that classify different flavours and ingredients, helping to understand the relationships between various flavours, such as their pairing or their common use in culinary contexts. Applying a cross-domain analysis to the Epic Kitchens dataset was useful for identifying the actions that recur most frequently across these categories.

4.3.4 CookDial Corpus

CookDial dialogue corpus [157] comprises 260 human-to-human English dialogues based on the culinary domain, in which an agent, given a recipe document extracted from the RISeC corpus [156], guides the user to prepare a meal. Data were collected by applying the experimental *Wizard-of-Oz* method [106], involving two participants interacting via a live chat platform. The application setup simulated the interaction between a voice assistant (agent) and a user. The agent had full access to the text of the recipe, while the user only knew its title. From this corpus, it was possible to identify relevant frames describing the processes of preparation of meals, analysing their distribution within the dialogue flow.

[11] FlavorDB: https://old.iiitd.ac.in/flavordb

4.4 Employed Tools

This section describes the tools employed to represent the domain elements to build the knowledge base required to conduct both the linguistic and probabilistic analyses described in Chapter 5. In this respect, an introduction to Knowledge Graphs for knowledge representation will be detailed, as they systematically organize facts, comprising entities, relationships, and semantic descriptions. Specifically, Neo4J, a non-native graph database, is designed to manage and store data in graph form, providing an effective means of structuring knowledge that both humans and machines can interpret.

4.4.1 Knowledge Graphs

Knowledge is essential for human existence and development, making the acquisition and representation of human knowledge fundamental tasks in AI research. Although humans naturally perceive and interpret their surroundings, AI systems require additional knowledge to achieve these abilities and deal with complex tasks in real-world scenarios [154, 242]. As seen so far, there has been the necessity of diverse approaches for representing human knowledge according to different conceptual models. In this regard, Knowledge Graphs have attracted significant attention from academia and industry as they systematically organise data in a graph format. Knowledge graphs have been the focus of research since the 2012 announcement of the Google Knowledge Graph [282] and have resulted in a wide variety of published descriptions and definitions, which remains still contentious [84, 31, 37]. In Hogan et al., [139], the *Knowledge Graph* (KG) is defined as:

"[A KG is] a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent potentially different relations between these entities" [p.71:3].

The term is often used interchangeably with *Knowledge Base* [77, 293, 92, 194, 93, 84, 58], even if there is a slight distinction. A Knowledge Base (KB) is a typical data set that represents real-world knowledge and semantic relations in triples, e.g. *head - relation - tail* or *subject - predicate - object* [38]. In this scenario, *Knowledge* specifically pertains to *explicit knowledge*, i.e., known information that can be explicitly written down [222, 139]. Entities can be real-world objects or abstract concepts,

while relationships illustrate the connections between these entities. The semantic descriptions associated with entities and their relationships encompass types and properties with clearly defined meanings [77, 139, 170, 242]. When the triplets are represented as a graph with edges as relations and nodes as entities, it is considered a Knowledge Graph (KG) [242]. In turn, A KB is itself often used as a synonym for *ontology*. As Ehrlinger [84] highlights, YAGO, which is considered an ontology according to its name, is also referred to both as a KB [77, 58] and as a KG [92, 305]. As mentioned earlier in Chapter 1, ontological representations enable semantic modelling of knowledge and are widely used as knowledge bases in AI applications, particularly in knowledge-based systems [84]. Applying an ontology as a knowledge base facilitates the validation of semantic relationships and the derivation of conclusions from known facts for inference, i.e., reasoning [95]. Although ontologies are sometimes mistakenly equated with database schemas, they include not only classes and properties (e.g., owl: ObjectProperty and owl: DatatypeProperty), but also instances (e.g., the ontology population) [86]. While the size of knowledge graphs is often cited as a defining characteristic and can be seen as large ontologies, some researchers [35] suggest that knowledge graphs possess additional characteristics that make them superior to ontologies [84]. The difference thus lies in the quantity (e.g., a large ontology), and the extended capabilities, i.e. the presence of an embedded reasoner that enables deriving new knowledge [84]. An example of KG is shown in Figure 4.3.

A key principle of any KG is to initially model data in graph form, as graphs provide a versatile method for conceptualizing, representing, and integrating varied and incomplete data [139]. At the state of the art, there are diverse typologies of graph data models. The most popular is *property graphs* as enables the association of nodes and edges with a collection of property–value pairs and labels, providing enhanced flexibility for data modelling [8, 209]. Property graphs are used in graph databases, offering a data management paradigm particularly suited to the construction and maintenance of KB. Its concept is based on the principles of graph theory, a mathematical framework for modelling relationships between objects [255]. Graphs are composed of nodes (also known as *vertices*), connected by edges representing their interconnections. The main characteristics and advantages of a graph database are here reported [254]:

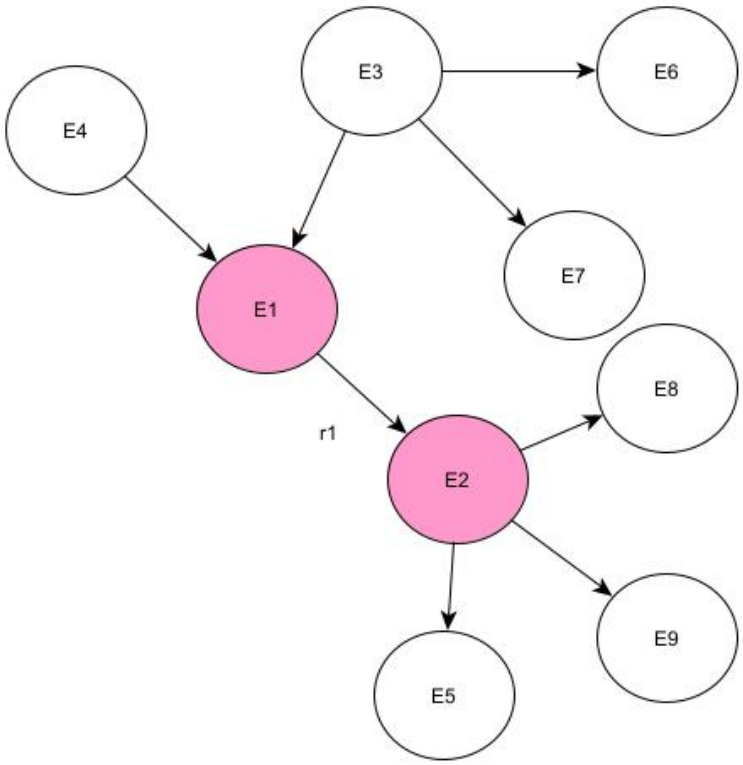

FIGURE 4.3: Example of a KG. Entities (*E1, E2, E3, etc.*) are represented as nodes linked through relationships (*edges*). In this context, *E1, R1, E2* is a triplet, indicating that E1 and E2 are connected by relation r1. (Source: Pengon et al., 2023 [242])

.

1. They are considered intuitive in that they organize information in a graph format, in which entities are represented as nodes and how they are connected through relationships;

2. Due to their graph structure, information retrieval is much more optimized compared to relational databases, as they leverage the proximity data of one or more roots (main nodes) of the graph database;

3. Those systems are flexible in their implementation compared to ontologies, as they leverage the proximity data of one or more roots within the graph database;

4. Diverse data types can be readily adapted, inserted, and connected, reflecting real-world cases.

One of the most popular property graph databases is Neo4J, an open-source graph database manager proven valuable for data representation [75], exploration [80] and visualisation [159].

4.4.2 Graph Databases: Neo4J

Neo4J is a NoSQL Graph Database Management System (GDBMS), i.e. software systems that enable data to be stored and organised without relying on the relational model typically used by traditional databases [134]. A classical example of a Relational Database Management System (RDMS) is MySQL[12], which organizes data in tables that are related to each other through predefined schemas and keys. Data in relational databases is typically structured in rows and columns, and queries are made using SQL (Structured Query Language) [202]. This model relies heavily on the relational model, where relationships between tables are established through foreign keys. In contrast to this model, Neo4J stores data in the form of nodes and relationships, allowing for a more flexible and efficient representation of complex, interconnected data. Indeed, nodes are used to represent entities and they can also be labelled with zero or more labels and can be linked to itself through relationship. As with nodes, relationships can have properties. A relation, which can only be of one type, connects two nodes and always has a starting node and an ending node. It is also important to note that it has a direction and can be traversed in both directions, which means that there is no need to add duplicate relations with opposite directions, as commonly happens for directed edge-labelled graph [139]. Cypher [105] is the native query language of Neo4j, allowing to create, modify, delete and query the database data. It is a graphical language, i.e. it is based on the graphical reproduction of the retrieved sub-graph and it is a declarative SQL-inspired language for describing visual patterns in graphs using ASCII-Art syntax[13]. Through Cypher, users can construct expressive and efficient queries to handle Create, Read, Update, and Delete (CRUD) functionality. An example of Cypher query on movie domain is shown in Listing 4.1. This query retrieves information about the directors and the movies they directed. The information in the brackets represents the nodes containing information about the entities. In this case, directors are represented by the PERSON node type and movies by the MOVIE node type. The square brackets contain information about the relationships between the nodes. In this case, the relationship type is DIRECTED, which indicates that the PERSON node (the director) directed the MOVIE node. The

[12]MySQL: https://dev.mysql.com/doc/ [last visited on 11/01/2025]

[13]ASCII art is a graphic design technique that uses computers for presentation. It consists of pictures pieced together from the 95 printable characters defined by the ASCII Standard from 1963, as well as ASCII-compliant character sets with proprietary extended characters. ASCII art can be created using any text editor and is often employed with free-form languages. Wikipedia: https://en.wikipedia.org/wiki/ASCII_art [last visited on: 11/01/2025]

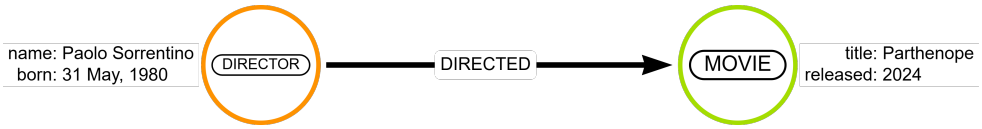

FIGURE 4.4: Graph representation of movie domain. It represents the node PERSON (orange) which is connected to the MOVIE node (green) through a relationship (DIRECTED). Each nodes are characterized by specific properties (e.g., the node PERSON shows two properties, such as "name" and "born").

use of the arrow (->) indicates that the relationship goes from the PERSON (the director) node to the MOVIE (the movie) node.

```
1 MATCH (p:PERSON)-[r:DIRECTED]->(m:MOVIE)
2 RETURN p, r, m LIMIT 1
```

LISTING 4.1: Cypher query example on the movie domain to retrieve information about the directors and the movies they directed. In this context, square brackets represent relationships (including their type), while normal brackets denote nodes. The hyphen dash indicates the direction of the relationship, with the arrow showing the flow from one node to another.

The MATCH command specifies the pattern to be searched in the graph, that is, a set of nodes and relationships that satisfy a certain condition. The RETURN command specifies what has to be returned as a result. Specifically, here is asked to return *p*, which is the node of the person (the director), *r*, which is the relation that connects the director to the movie (the DIRECTED relation) and *m*, the node of the movie (the movie that the director directed). Lastly, LIMIT 1 limits the number of results to 1. This means that only a single result (a single director and the movie he directed) will be returned, even if there are many directors and movies that satisfy the query pattern. Thus, the result will contain for the director, the relationship they have with the movie and the movie itself, as shown in Figure 4.4.

An important concept in Cypher Language is the ability to parameterize queries. There may be situations where multiple identical queries need to be executed, differing only in the values assigned to certain properties. Suppose we have a Movie database where each movie node features properties such as `title`, `genre`, and `releaseYear`. Rather than composing multiple queries for distinct genres and years, it is possible to employ parameters. For each query where the property is explicitly specified, Neo4j calculates a distinct execution plan, which varies across queries.

Conversely, parameterizing a query enables Neo4j to define a single execution plan, which is then retained for subsequent executions of the same query, as shown in Listing 4.2, which includes a parameter for the `Person.name` property.

```
MATCH (p:Person {name: $name})-[:ACTED_IN]->(m:Movie)
```

LISTING 4.2: This query retrieves all the films `(m:Movie)` in which a person `(p:Person)` with a specific name (`$name`) has starred. The string `(p:Person {name: $name})` searches for a node with the label `Person` that has an attribute `name` equal to `$name` (a parameter passed to the query).

As knowledge is formalized through a graph, a key functionality of Neo4j is the search for paths within the graph [14]. The Path is a specialized data type that represents the graph structure, capturing the structure of nodes and relationships as a result of queries, which can be obtained as shown in Listing 4.3.

```
MATCH p = (:Person {name: "Tom Hanks"})-[:ACTED_IN]->(m:Movie
    )
RETURN p
```

LISTING 4.3: `length(p)` returns the number of relationships in the path, `nodes(p)` returns the array of nodes in the path in the order they are traversed, and `relationships(p)` returns the array of relationships in the path in the order they are traversed.

If it is intended to get from one particular node to another, but the path in the graph that enables it to do so is not known, it is possible to specify patterns such as `()-[*]-()`, with a star * placed on the relationship. A single star indicates `*1..`, and the complete syntax of the operator is ⟨`min .. max`⟩, with $min, max \in \mathbb{N}_0$, as reported in Listing 4.4.

```
MATCH p = (a:Person)-[:IS_FRIEND_OF*0..1]->(b:Person)
WHERE a.name = "Al" AND b.name = "John"
RETURN p
```

LISTING 4.4: This query finds paths between two people, `Al` and `John`, using at most one `IS_FRIEND_OF` relationship. The `*0..1` operator allows zero or one hop in the friendship connection.

[14] Neo4J Blog: https://neo4j.com/blog/developer/the-power-of-the-path-1/ [last visited on: 24/03/2025]

Dialogue-based applications are often based on knowledge graphs, as they are efficient in representation for deepening the relationships between how people use the domain knowledge in human-human dialogues. This supports the modelling of how a machine should mimic human behaviour. Explainable-by-design systems do not rely solely on machine learning to surface these strategies, so it is important to cross-reference recorded dialogues with the underlying knowledge concepts they employ. Neo4j provides excellent integration with Machine Learning algorithms, which can be useful for extracting implicit knowledge [114]. The model proposed in this work leverages graph databases to implement deductive capabilities through path-based queries, thereby recapturing some of the inferential capabilities of traditional inference engines, but with the performance advantages of modern database technology. Given the extensive mathematical theory about graphs, they represent an ideal way of representing knowledge that is interpretable for both humans and machines. Their ability to naturally represent relationships between information, their flexibility and query efficiency make them indispensable tools for the development of intelligent applications, such as conversational agents. Specifically, their intuitive data representation supports the detection of specific regularities within the network in the form of recurrent sub-structures. Following the approach adopted in Di Maro [74] and Origlia et al., [229, 228], graph databases can be used to represent knowledge about the domain and how it is used in reference dialogues for linguistic studies simultaneously. Specifically, as highlighted in Di Maro's work [72], graphs have proven to be an efficient tool for corpus analysis. Different sets of shared knowledge were represented as graphs, enabling a deeper understanding of dialogue phenomena and supporting more informed design of dialogue systems.

4.5 Conclusions

This chapter has outlined the concept of conceiving CSK as a dynamic process that emerges and evolves through communication, rather than seeing it as a static repository of information. Knowing what can be taken for granted and what needs to be made explicit means demonstrating communicative competence, which is still a challenge for conversational agents. Given the large amount of implicit information widely assumed by humans and the impossibility of representing it fully and exhaustively in a single resource, CSK is not necessarily confined to explicit representations

of knowledge relations but can emerge through communicative interaction and be reconstructed through entities that show meaningful correlations with high probability. To conduct the analysis, the cooking domain is taken into account for (i) its ubiquitous familiarity and (ii) the presence of implicit systematic chains of actions in the cooking process. Information are divided into three macro categories, allowing for a more structured approach to managing information, thereby facilitating a clearer understanding and more effective analysis of the data. (i) Foreground knowledge is defined as information explicitly expressed in both oral communication and written texts; (ii) Background knowledge pertains to basic fundamental information about entities often left omitted in communication. Lastly, (iii) Presupposed Knowledge is categorised as the implicit information automatically inferred by speakers. A three-level analysis is proposed to uncover the processes underlying the foreground information. The I Level pertains to the action and the entity involved in a foreground event; the II Level relates to the comprehensive knowledge about entities and actions. The III Level focus on the presupposed action and entities that enable the frame identified in the II level to take place. In the next chapter, each of these three levels will be analysed in detail.

CHAPTER

FIVE

INVESTIGATING DATA EMPLOYING THE THREE-LEVEL ANALYSIS APPROACH

This chapter discusses the analyses conducted across the three levels of analysis presented in Chapter 4.

5.1 I Level: Semantic Analysis

This section outlines the analysis conducted across the three levels of analysis depicted in Chapter 4. Specifically, the focus is on the first level of analysis, which is addressed by examining the dialogical corpus CookDial. Leveraging the FrameNet lexical base, semantic domain information is identified to extract the positions of explicit information within the dialogue flow, offering an overview of action distribution and answering RQ1.

5.1.1 CookDial's Information Distribution Analysis

This section details the analysis conducted by Mennella et al. [205] on the semantic characteristics of foreground information based on Frame Semantics theory [101]. For the development of the annotation scheme, the FrameNet lexical database [16] is employed, from which the domain frames are identified. The corpus is annotated using Label Studio, an open-source tool ideal for performing linguistic tasks of this type, annotating 46 dialogues out of a total of 260 contained in the corpus. The section concludes by presenting the results obtained from the analysis pertaining to the information distribution along the dialogue flow, establishing which actions are mostly involved in the initial and final stages of the preparation of a dish.

5.1.2 Methodology

In Frame Semantics Theory, a semantic frame is defined as a coherent structure of concepts which evokes a situation, an event or a state along with its participants. In FrameNet, each concept (*frame*) is schematized with its definition, its examples, and its *frame elements*, which represents the semantic roles required by the *lexical unit* (LU) evoking the frame. Each FE can be (*core*) or (*non-core*), depending on the frame in which they appear, necessary for expressing specific meanings of the frame of the argument. Taking into account the sentence *Bake the cookies at 350 degrees*, it evokes *Apply_heat* frame described as follows:

Frame Description

A Cook applies heat to Food, where the Temperature_setting of the heat and Duration of application may be specified. A Heating_instrument, generally indicated by a locative phrase, may also be expressed.

Elements such as *Cook*, *Food*, *Temperature_setting*, *Heating_instrument*, *Duration* are the FEs of the frame. verbs such as *fry, bake or boil* represent the LUs evoking the frame. In this work, *frames* are labelled as *Frame Intent (FIs)* as they determine the explicit actions expressed by users. As far as *Frame Elements (FEs)* are concerned, the nomenclature remains the same as in FrameNet.

In this work, 29 domain-based FIs and their corresponding FEs were identified, as reported in Table 5.1. A complete list of frame description can be found in Appendix 7.

FRAME INTENT (FI)	EXAMPLE	FRAME ELEMENT (FE)
Apply_heat	Bake the cookies at 350 degrees	Temperature_setting
	Fry the doughnuts in the deep-fryer	Heating_instrument
	Bake the cake for 10 minutes	Food
	Bake the cake for 10 minutes / until it becomes golden brown	Duration
	Fry the chicken in oil	Medium
	Bake the dough on the top shelf of the oven	Place
	Bake onion with oil	Co-participant

TABLE 5.1: Annotation of Frame Intents (FIs, e.g. *Apply_heat*) and Frame Elements (FEs) along with its examples.

After the annotation scheme was established, the CookDial dialogue corpus was taken into account for domain frame annotation. The entire corpus, composed of 260 dialogues, was loaded in Label Studio [302], an open-source data labelling platform particularly useful for linguistic analysis as it offers customisable interfaces

FIGURE 5.1: Label Studio interface. The upper section displays the labels for FIs and FEs, while the lower section presents the full dialogue. Highlighted text segments within the dialogue correspond to the assigned labels.

for various data types. Its innovative design is particularly effective for linguistics task annotation, such as semantic meaning, thus supporting linguistic research and machine learning training. The dialogues were imported into the platform using a Python script (Appendix C), which extracts and reorganises the original content in a different format to facilitate the annotation. In this respect, each dialogue of the corpus consists of turns that reflect the interactions between an User and a Bot, as in the original study were employed three paid workers who conversed in pairs via a web chat platform, alternating between user and agent roles after each conversation. Consequently, for each dialogue, the script loads the dialogue rounds, identifies whether they were written by the 'Bot' or the 'Human' (user) by labelling them and creates a new JSON file with a structure that also includes the dialogue ID. It is important to note that each dialogue corresponds to a recipe contained in RISeC corpus, which the bot employs to guide the user in preparing the target meal. Therefore, in Label Studio, the shifts of each dialogue turn are labelled as *Bot* and *Human*, corresponding to the utterances elicited by either the agent or the user, as illustrated in Figure 5.1. The upper section displays the FIs and FEs labels, while the lower section shows the entire dialogue. For this work, only the Bot's frames sentences were taken into account, labelling 46 out of the 260 dialogues by a single annotator. Since Label Studio allows multiple users to collaborate on the same project, it has enabled the evaluation of annotation agreement, allowing a second annotator [1] to label the first 10 dialogues to verify the validity of the proposed model. An example

[1]In this respect, I would like to thank Domenico Morra, a student enrolled in the Master's Course in Foreign Languages at the University of Naples Federico II, for his help in this matter.

annotation is provided below, which presents a dialogue along with its corresponding recipe instruction. The annotation showcases the frames in the order they appear along the dialogic flow. The full frames annotation for the first 10 dialogues can be found in the Appendix B.

English Walnut Pie - Instructions (RiSeC)

- Preheat oven to 400 degrees F (205 degrees C).
- Beat the eggs in a large bowl.
- Mix in sugar, salt, vanilla, and syrup.
- Melt the butter or margarine, and add it to the egg mixture.
- Stir in the nuts.
- Pour filling into pie shell.
- Bake in preheated oven for 10 minutes.
- Reduce heat to 300 degrees F (150 degrees C), and continue baking for 35 to 45 minutes.

Annotation (CookDial). English Walnut Pie - Dialogue ID: 0

- **Cause_temperature_change**. Item: oven; Temperature_Setting: 400 degrees Fahrenheit, 205 degrees Celsius
- **Taking**. **Theme**: bowl; **Cause_to_amalgamate**. Parts: eggs
- **Cause_to_amalgamate**. Parts: white sugar, vanilla extract, light corn syrup
- **cause_change_of_phase**. Patient: butter, margarine. **Cause_to_amalgamate**. New_member: it; Existing_member: mixture
- **Cause_to_amalgamate**. Parts: chopped walnuts. **Cause_to_be_included**, New_member: them; Existing_Member: mixture
- **Mass_motion**. Mass_Theme: filling; Goal: pie shell
- **Apply_Heat**. Food: cake; Duration: 10 minutes. **Cause_temperature_change**. Item: oven; Temperature_Goal: 150 degrees Celsius. **Cause_to_continue**. State: cake; Duration: 35 tot 45 minutes.
- **Cause_temperature_change**. Item: oven; Temperature_Goal: 150 degrees Celsius. **Apply_Heat**. Duration: 35 to 45 minutes

5.1.3 Results

Following the completion of the annotation phase, the dialogues were exported from the Label Studio and through a Python script (Appendix C), the MASI (Measuring Agreement on Set-valued Items) distance [238] was employed to ensure the annotation agreement. This distance is particularly useful for handling multiple labels for a single item, ranging from 1 to indicate identical sets and to 0 to indicate completely disjointed sets. Employing MASI, it is possible to quantify the level of agreement between annotators, thus improving the reliability of the annotation process. Additionally, Krippendorff's Alpha [237] was applied to assess the annotation quality, calculating the metric of weighted agreement. This metric is particularly effective

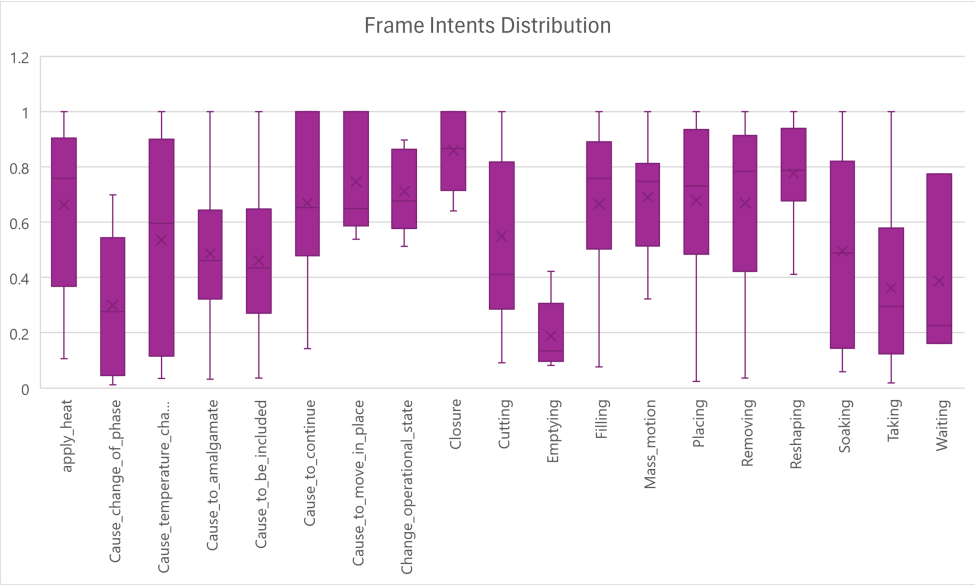

FIGURE 5.2: FI distribution within the dialogues. Only 19 out of 29 FI are taken into account for the analysis.

for assessing inter-rater reliability in multi-annotator contexts, as it accounts for the degree of agreement beyond what would be expected by chance. By incorporating weights, Krippendorff's alpha allows for a nuanced assessment of the agreement, reflecting the importance of different annotations and improving the overall assessment of annotation consistency and quality. This approach ensures that the results derived from the annotated data are reliable, showing an agreement value of 0.75, thus confirming the validity of the annotation scheme. To ascertain the FI's distribution within the dialogue stream, a Python script was executed (Appendix C.3). The code analyses annotated dialogues to calculate the normalized distribution of FIs in the dialogue turns. For each annotation, it checks whether the FI is in the list, records the shift in which it appears, updates the maximum shift if necessary, and normalizes the shifts against the maximum. The normalized data from each dialogue is gathered into a single list and converted into a Pandas DataFrame for tabular analysis. For enhanced visualisation purposes, the data was then converted into a graphical representation, as illustrated in Figure (5.2).

From 29 FIs, it has been identified 19 relevant for the analysis. Results show that certain FI as *Taking* (e.g. take a bowl), *Soaking* (e.g. soak the chicken), *Emptying* (e.g. drain the turkey), *Cause_temperature_change* (e.g. preheat the oven to 400 degrees), and *Cause_change_of_phase* (e.g. melt 1/4 cup butter) occur earlier, while *Cause_to _continue* (e.g. keep the chicken warm), *Cause_to_move_in_place* (e.g. turn the pancake), *Reshaping* (e.g. roll up each crepes), *Placing* (e.g. put

the chicken on plate) and *Closure* (e.g. seal the bag) occur towards the process's end. This distribution reflects the natural flow of a culinary task, where initial steps involve preparing ingredients (Taking, Soaking, Emptying) and manipulating temperature (Cause_temperature_change, Cause_change_of_phase), while later stages focus on cooking food (Cause_to_move _in_place), monitoring progress (Cause_to_continue), modelling the shape (Reshaping) and finalising the process (Placing, Closure).

5.1.4 Discussion

In this section, linguistic analysis has been applied to deepen the investigation of semantic features, examining the CookDial corpus. Leveraging the FrameNet lexical base, semantic domain information has been identified to extract the positions of explicit information from CookDial within the dialogue flow. An annotation scheme has been established, identifying 29 domain-based FI along with their FE. Dialogues were loaded in Label Studio and 46 out of the 260 dialogues were manually annotated by a single annotator. Additionally, the first 10 dialogues were annotated by a second annotator to verify the validity of the proposed annotation model. Employing MASI and Krippendorff's Alpha measures, was possible to assess the annotation quality, calculating the metric of weighted agreement. Results show an agreement value of 0.75, confirming the validity of the annotation scheme. To ascertain the FI's distribution within the dialogue stream, a Python script was executed. From 29 FIs, it has been identified 19 relevant for the analysis. Results suggest that certain FI occur earlier in the dialogue, while others occur towards the process's end. This distribution reflects the natural flow of a culinary task, where initial steps involve preparing ingredients (Taking, Soaking, Emptying) and manipulating temperature (Cause_temperature_change, Cause_change_of_phase), while later stages focus on cooking food (Cause_to_move_in_place), monitoring progress (Cause_to_continue), modelling the shape (Reshaping) and finalising the process (Placing, Closure), offering an overview of actions performed before others. The analyses conducted so far highlight results that, at first glance, may seem quite trivial. However, it is precisely because of this triviality that these results prove to be important for CS. Analyzing the position of the evoked frames can reveal a preferred execution pattern, regardless of the dish being prepared. In other words, the identification of frames highlights that

there are steps in the process of preparing a dish that are held to be crucial (and this is explains why those information are made explicit), while the distribution shows that they are stated in a dialogue respecting a precise logical order. In this respect, it can be stated that RQ1 has been satisfied. After having identified the explicit semantic frames and examined the order in which actions are mentioned in the process of preparing a dish, we can now improve our understanding of this data by analyzing which actions recur most frequently on certain categories of objects, as referred in RQ2.

5.2 II Level: Domain Representation

This section outlines the construction of the knowledge base by integrating three resources concerning flavours, recipes and instructions into the graph database. Data provided by Recipe1M+, FlavorDB and Epic Kitchens were first imported into the database and then interconnected. The frequency of actions extracted from Epic Kitchens performed on ingredients belonging to specific categories within FlavorDB has been quantified, providing insights into the categorical distribution of ingredients on which actions are performed.

5.2.1 Building the Domain Knowledge Graph

To build the domain knowledge in Neo4J, the first step of the graph-building procedure consists of importing the data provided by resources described in Chapter 4 to support a cross-resource analysis. Specifically, Recipe1M+, FlavorDB and Epic Kitchens are taken into account as they are the most structured data sources available for the specific domain. Those sources represent the domain knowledge of ingredients, recipe titles with instructions, food flavours and actions performed in a kitchen. They were first imported independently into Neo4J, with their specific nodes and relation structure, represented as separate sub-graphs. To connect the Recipe1M+ and FlavorDB, a query (Listing 5.1) was implemented to link nodes with similar properties. The full-text mechanism was used to identify flavour aliases for each ingredient. The system automatically created relationships labelled as SAME_AS between ingredients' properties and their most relevant flavour aliases, as determined by a relevance score associated with each search result. To assess the accuracy of this

automated linking process, a manual analysis of a representative sample of 300 entities was conducted, revealing only 39 instances of incorrect linking, thus suggesting a high degree of precision in this approach. A simplified graphical representation is shown in Figure 5.3. The purple node represents a recipe, which is linked to the INSTRUCTION node and INGREDIENT node through HAS_INSTRUCTION and HAS_INGREDIENT relationships, respectively. The INGREDIENT node is connected to the FLAVORDB node through the SAME_AS relation which is, in turn, linked to a series of MOLECULE nodes representing the flavour's constituent molecules.

```
MATCH (i:INGREDIENT&Recipe1M)
WHERE i.primaryName IS NOT NULL
WITH i
CALL { WITH i
    CALL db.index.fulltext.queryNodes("FlavorDBAliases",
     apoc.text.regreplace(i.primaryName, "[^a-zA-Z0-9 ]", ""))
     YIELD node, score
     WHERE "FLAVOR" IN labels(node)
    RETURN node, score ORDER BY score DESC LIMIT 1
      }
WITH i, node, score
MERGE (i)-[:SAME_AS]->(node)
```

Listing 5.1: Cypher query employed to connect ingredients from Recipe1M+ to node flavours from FlavorDB

The Epic Kitchens corpus does not refer to specific recipes but consists of generic actions related to cooking processes and everyday activities performed in a kitchen. Within each verb category, synonyms denoting the same core action were grouped. For instance, the verb class *6* encompasses all synonyms that convey the concept of mixing ingredients, including terms such as *stir*, *mix* and *whisk*. To distinguish *core* actions, i.e., actions that are mentioned explicitly in a cooking instruction (e.g., *whisk eggs*), from *non-core* actions, i.e., a set of actions that are never mentioned explicitly but are performed anyway (e.g., *break eggs*), each sentence in the dataset was manually annotated by one human annotator employing the labels ACTION and INSTRUCTION. The INSTRUCTION label indicates core actions, while the label ACTION indicates all non-core actions (more detail will be provided in next section

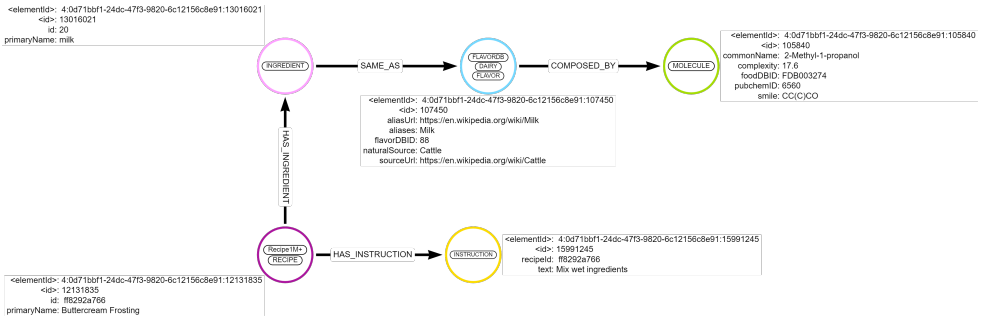

FIGURE 5.3: Graphical representation of the interconnection between Recipe1M+ and FlavorDB resources. The purple node represents a recipe (*Buttercream Frosting*). This recipe is linked to the INSTRUCTION node (yellow) and INGREDIENT node (pink) through HAS_INSTRUCTION and HAS_INGREDIENT relationships, respectively. The latter, representing the ingredient *milk*, is linked to the FLAVORDB node (blue) through the SAME_AS relation. The FLAVORDB node is in turn linked to a series of MOLECULE nodes (green), representing the flavour's constituent molecules. Here, only the molecule *2-Methyl-1-propanol* is shown. It is important to note that this representation leveraged existing entities within the database.

5.3). For practical purposes, these sequences are stored in the form of a directed acyclic graph, where each node represents an action and has exactly one incoming and one outgoing arc - except for the initial and end nodes, which contain zero arches). The arcs, stored in the Neo4j as FOLLOWED_BY relationships, represent the sequence of actions that make up each of them. As already highlighted, the dataset does not contain structured recipes like those found in Recipe1M+, but rather a list of generic actions carried out in the kitchen. Each participant in the original dataset had an ID along with their records, which, in turn, had an ID based on the day the recording was made. Since specific instructions were not followed and each record reported more than 100 actions, a brief description of the actions contained in the videos, such as "cooking rice" or "making an omelette," is provided. For this reason, each group of those actions is referred to as a *recipe* for simplification purposes and labeled as EPIC_KITCHEN to differentiate them from the rest of the information contained in the database. Each of these nodes contains three properties: (i) *recipeID*, which represents the identifier of the recipe to which the action belongs; (ii) *id*, the position of the action in the recipe; (iii) *narration*, which consists of the full description of the action. The EPIC_KITCHEN nodes are also individually connected to NOUN nodes through the HAS_NOUN relationship, storing information about the main noun of the narration, and to VERB nodes through the HAS_VERB relationship, storing information about the main verb of the narration, as shown in Figure 5.4. NOUN and VERB nodes have respectively *noun* and *verb* properties.

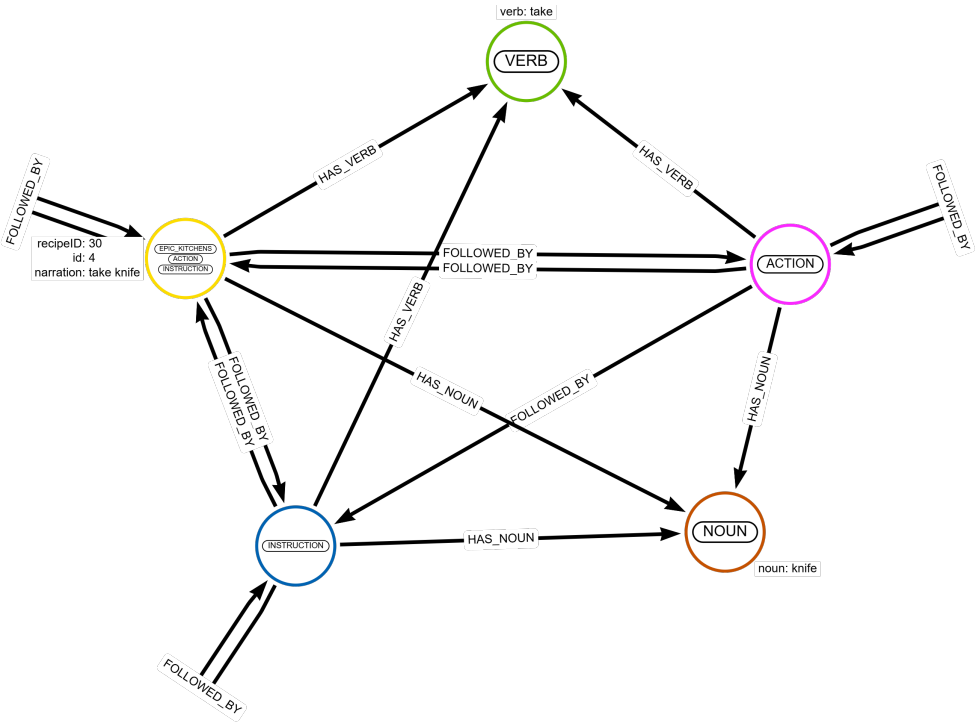

FIGURE 5.4: The property narration *take knife* has an EPIC_KITCHEN node, labelled as ACTION|INSTRUCTION, linked to the VERB node with a property verb *take* and to the NOUN node with a property noun *knife*,

Cross-resource relationships can be found using dedicated indexes. Indeed, in Neo4j, a full-text index using the Lucene query engine allows performing fuzzy searches over string properties. This means that it can cover multiple node properties, allowing searches among various textual attributes such as names, descriptions and instructions. This versatility allows users to find information related to different aspects of the data. In this work, full-text indexes were created in the database to index the properties containing item names such as recipes and ingredients. Through the use of Query Cypher (Listing 5.2), it was possible to extract all the instructions found both in the Recipe1M+ and the Epic Kitchens dataset for the same ingredient, also considering all possible aliases of that ingredient, as reported in the FlavorDB database.

```
1 MATCH (ins1:INSTRUCTION)<-[:FOLLOWED_BY*0..]-
2         (:INSTRUCTION)<-[:HAS_INSTRUCTION]-
3         (rec:RECIPE&Recipe1M)-[:HAS_INGREDIENT]->(ing:
              INGREDIENT)
4 WHERE ing.primaryName IN ins1.text
5 WITH ing.primaryName AS ingredientName,
```

```
6       COLLECT(DISTINCT ins1.text) AS instructions1 LIMIT 10
7   CALL db.index.fulltext.queryNodes("FlavorDBAliases",
        ingredientName)
8   YIELD node, score
9   WITH instructions1, node.aliases AS FlavorDBAliases,
        ingredientName
10  UNWIND FlavorDBAliases AS flavorDBAlias
11  CALL db.index.fulltext.queryNodes("epicKitchenNames",
        flavorDBAlias)
12  YIELD node, score
13  WITH instructions1, node AS EpicKitchensNode, flavorDBAlias,
        ingredientName
14  MATCH (EpicKitchensNode)<-[:HAS_NOUN]-(ins2:INSTRUCTION)
15  WITH ingredientName, instructions1,
16       COLLECT(DISTINCT ins2.narration) AS instructions2
17  RETURN ingredientName, instructions1 + instructions2 AS
        instructions
```

Listing 5.2: Query Cypher to extract recipe instructions mentioning an ingredient and enrich them with data from FlavorDB and Epic Kitchens.

The query first explores recipes belonging to the Recipe1M dataset and extracts, for each recipe, the set of instructions naming one of the ingredients used in the recipe (lines 1 and 2). Instructions are collected in a single list and passed on together with the target ingredient name (line 3). Results are limited to 10 ingredients for efficiency reasons in this example. The name of the ingredient is used to find, in FlavorDB, the nodes containing a name similar to the target ingredient's one. This search is performed using a specialised fulltext index (line 4). The results are passed on (line 5) and, for each ingredient alias (line 6), corresponding ingredients are searched for in Epic Kitchens using another fulltext index (line 7). Results are passed on (line 8) and, for each matching ingredient, the explicit pattern linking ingredients to instructions in Epic Kitchen, which is different than the one found in Recipe1M, is used (line 9) to extract instructions using that ingredient. The two lists of instructions are then merged (line 11) to obtain the final list of instructions using the same ingredient in two different datasets, also considering aliases. An example of ingredients along with their instruction is reported in Table 5.2.

ingredient_Name	instructions
Parsley	press parsley, wash parsley, put parsley in tray, [...]
Cinnamon	pour in cinnamon in bowl, mix in cinnamon, spoon in cinnamon
Butter	spread butter, spread butter on bread, put butter on frying pan
Salt	put salt to the pasta, sprinkle salt onto bowl, pour salt, [...]
Milk	pour the milk into the bowl, pour milk into cup, pour milk in cup, [...]
Flour	mix the flour with the fish, spread flour on counter, add flour to mix, [...]
Water	pour water into cup, pour water in the pasta, empty water in tray, [...]

TABLE 5.2: Ingredients along with their instructions. Due to the complexity and breadth of the lists, only a small part of the ingredients are shown in this table.

5.2.2 Cross-Domain Analysis

As we have seen so far, Neo4j employs a full-text index using the Lucene query engine to perform fuzzy searches over string properties. This allows it to cover multiple node properties, enabling searches among various textual attributes such as names, descriptions, and instructions. This way, users can easily find information related to different aspects of the data. Therefore, a cross-resource analysis can be performed using dedicated indexes. In this respect, to answer RQ2, the frequency of Epic-Kitchens actions performed on ingredients belonging to specific categories within FlavorDB has been quantified to glean insights into the categorical distribution of ingredients on which actions are performed.

5.2.3 Methodology

Given the extensive size of the Epic Kitchens dataset, it has opted to present the inherent co-occurrence patterns for selecting one noun, in this case *onion*, as shown in Figure 5.5. To ensure clarity and focus on the most common actions performed on this object, it has been taken into account only occurrences that exceeded the threshold of 10. As the results show, the actions associated with *cutting*, *peeling* and *stirring* have the highest frequency, while those related to object manipulation, such as *moving* and *placing* (e.g. an object on a surface), show the lowest occurrence rates. Starting with the three most relevant actions (cutting, peeling and stirring) on this noun, it has been performed a cross-analysis on the database, extracting the instance of each action performed on ingredients belonging to designated categories in FlavorDB.

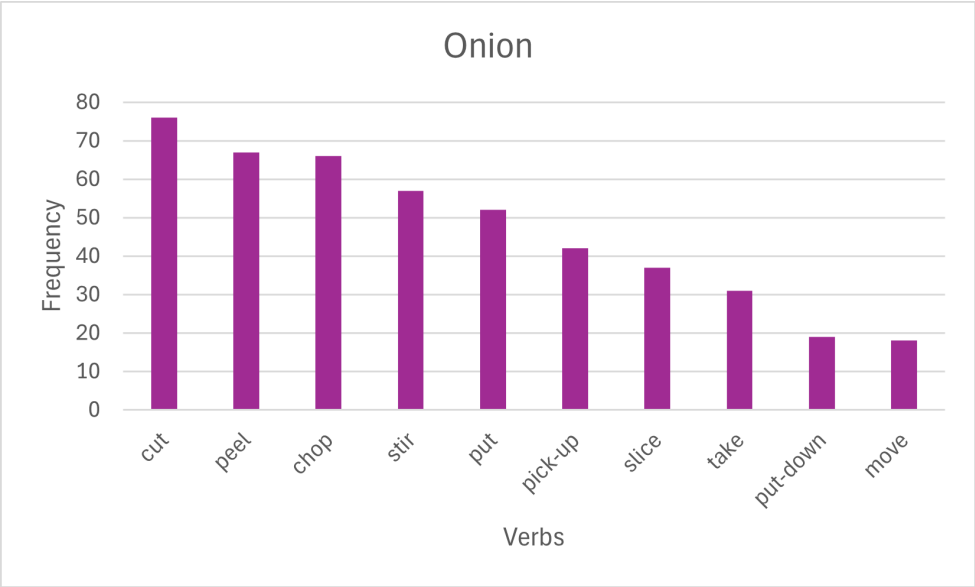

FIGURE 5.5: Co-occurrence of verbs patterns on the noun *onion*.

```
MATCH (v:Verb_Class)<-[:INSTANCE_OF]-(vi:Verb_Class_Instance)
MATCH p= shortestPath((vi)<-[:NEXT*]-(i:MC_Start))
WITH v, last(nodes(p)) AS i
MATCH (i)-[:SAME_AS]->(f:FLAVOR)
RETURN toInteger(v.verb_class) AS VerbClass,
apoc.coll.subtract(labels(f), ['FLAVOR', 'FLAVORDB'])[0]
AS FlavorData, COUNT(*) AS Count
ORDER BY VerbClass ASC, Count DESC
```

LISTING 5.3: Query employed to extract the frequency of action performed on ingredients belonging to FlavorDB categories.

Employing a Query Cypher (Listing 5.3), it has been estimated the frequency of actions performed on specific ingredient categories. It is noteworthy that values less than one were excluded from the analysis, except for those pertaining to the *peeling* action, as shown in Figure 5.7.

5.2.4 Results

The analysis revealed that the verb *cut* (Figure 5.6) is particularly prevalent for ingredients classified as Fruits, Vegetables and Spices and less prevalent in Fruits_citrus and Vegetable_cabbage categories. It is important to note that certain categories within FlavorDB exhibit a granular level of detail. For example, the Fruit_citrus category exclusively encompasses ingredients classified specifically as citrus fruits. Similarly,

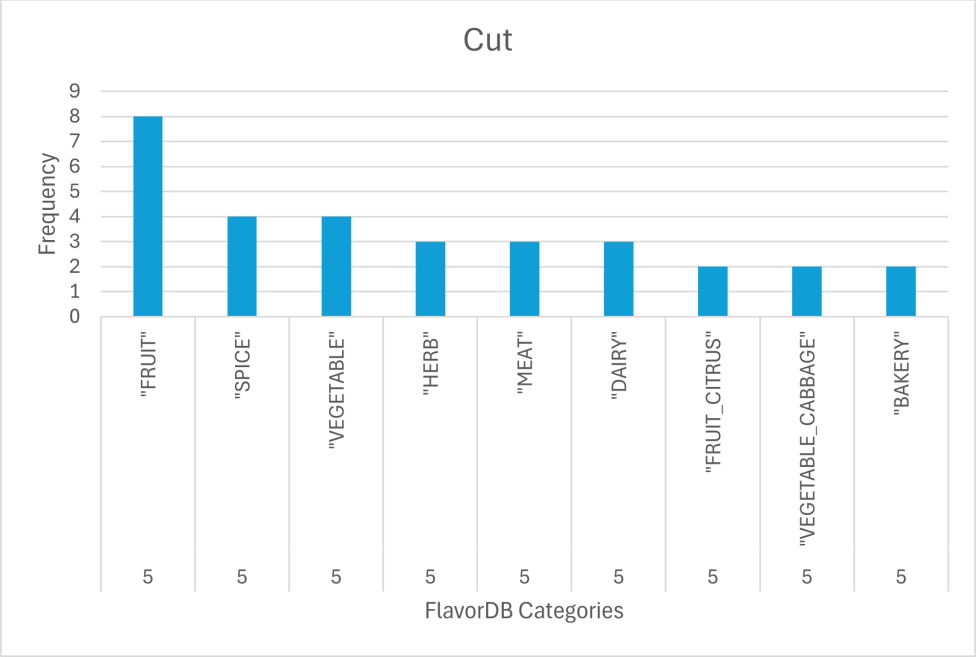

FIGURE 5.6: Frequency distribution for verb class *cut* in FlavorDB Categories. The numbers on the horizontal axis indicate the verb classification in Epic Kitchens.

the Vegetable_cabbage category contains subcategories dedicated to specific vegetable families, such as the *cabbage family*. Similar distinctions are applied to the remaining categories analysed in the following results. Concerning the peeling action in Figure (5.7), it can be seen that the Meat category is completely absent, while the Fruit category seems to be the prevalent one. Subcategories like Vegetable_root and Vegetable_tuber do not demonstrate a particular frequency with this action. Lastly, categories presenting the *stir* verb (Figure 5.8) concern primarily the Bakery category, followed by Additive, Herb and Vegetable. The Bakery category consists of ingredients related to processed products (pizza, cakes, etc.). The Addition category includes ingredients typically added to enhance food and beverages (e.g. margarine, sugar, etc.). Conversely, a lower frequency of co-occurrence is observed for ingredients categorised as Fruit, Meat, Cereal, Fish_seafood, and Beverage, suggesting potential distinctions in how these ingredient categories are typically processed.

The observed co-occurrence patterns between specific actions and certain ingredient categories suggest that particular actions are frequently associated with specific ingredient categories. In essence, the inherent association between an action and an ingredient category is well-established within the knowledge base.

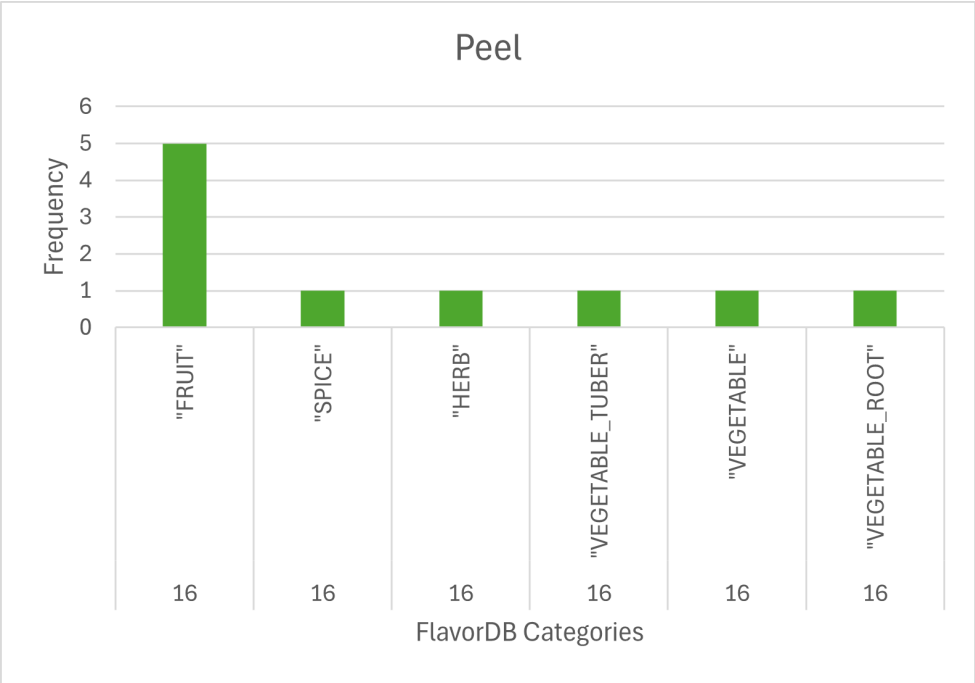

FIGURE 5.7: Frequency distribution for verb class *peel* in FlavorDB Categories.

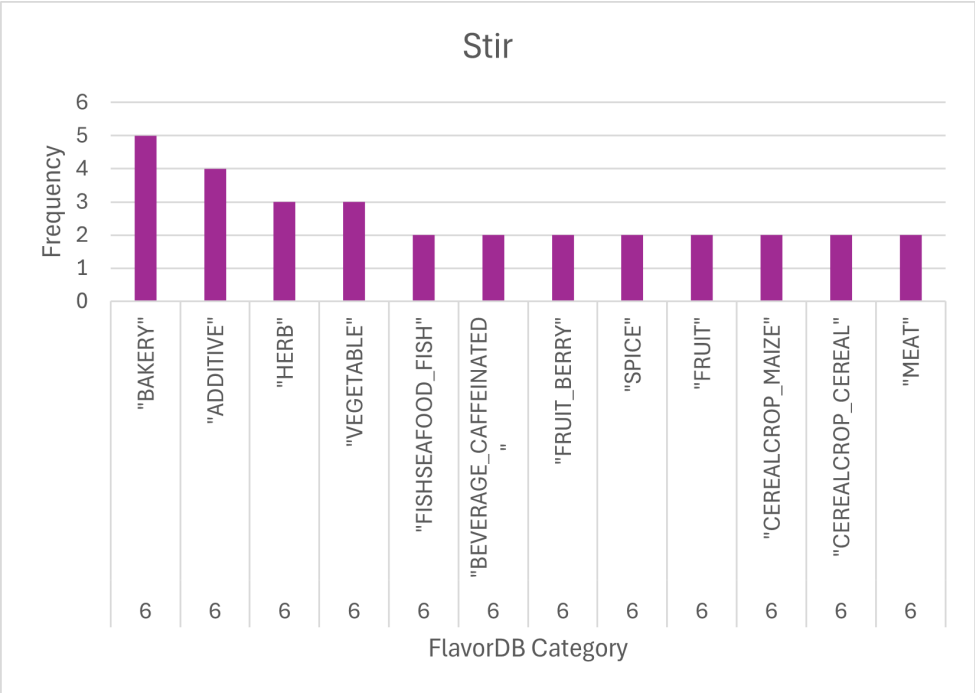

FIGURE 5.8: Frequency distribution for verb class *stir* in FlavorDB Categories.

5.2.5 Discussion

In this section it has been detailed how the cooking resources were integrated and represented within the graph database, thereby establishing the domain knowledge base. Data provided by Recipe1M+, FlavorDB and Epic Kitchens were first imported independently into Neo4J, with their specific nodes and relation structure and then interconnected. Subsequently, it has been quantified the frequency of Epic Kitchen actions performed on ingredients belonging to specific categories within FlavorDB, gleaning insights into the categorical distribution of ingredients on which actions are performed. Three main verbs were investigated: *cut*, *peel* and *stir*. The analysis revealed that the verb *cut* is particularly prevalent for ingredients classified as Fruits, Vegetables and Spices and less prevalent in Fruits_citrus and Vegetable_cabbage categories. Which concerns the verb *peel*, the Meat category is completely absent, while the Fruit category seems to be the prevalent one. Subcategories like Vegetable_root and Vegetable_tuber do not demonstrate a particular frequency with this action. Lastly, categories presenting *stir* concern primarily the Bakery category, followed by Additive, Herb and Vegetable.

The results obtained indicate that there is a correlation between the object and its manipulation. For instance, we have observed that fruits and vegetables are frequently associated with the *cutting* action. However, the *peeling* action exhibits a stronger association with fruits compared to vegetables. Conversely, vegetables demonstrate a high frequency with *stirring* actions compared to fruits. Instead, for the category Meat, we saw that it only appears associated with the verb *cut*. In other words, fruit and vegetable categories share a similar frequency distribution between the actions of 'cutting' and 'peeling' as they likely share structural similarities, despite a greater preference for one of the two. For instance, both apples and carrots are commonly cut into pieces for cooking or eating. The action of cutting them typically involves slicing or chopping them into smaller sections. Apples and carrots can also be peeled, although it is more common for carrots to be peeled before eating, while apples may be peeled depending on personal preference or recipe. Therefore, the similarity in structure (e.g. olid, cylindrical/round) makes both the apple (fruit) and the carrot (vegetable) likely to experience a similar frequency of being cut and peeled during preparation. In contrast, meat category is completely excluded to find actions inherent to 'peeling' as the action of peeling is generally not applied to meat. For instance, chicken is commonly cut into pieces for cooking or preparation, often chopped into

smaller parts like breasts, thighs, or wings. Even if chicken is usually prepared in a deboned or skinless manner, the action of peeling, i.e. removing a layer of skin in the way fruits or vegetables are peeled, does not apply to meat in the same way. Therefore, it can be stated that the frequency of these actions demonstrates a strong relationship between the object and the types of manipulation applied to them, answering RQ2. As this information is well established from experience, it can be omitted from communicative exchanges, as it belongs to the general understanding of the world. Therefore, the recovery of implicit information (i.e. what has been classified as *Presupposed Knowledge* in Chapter 4) is possible by retrieving it through a probability estimation within the graph, discussed in the next section.

5.3 III Level: Probabilistic Analysis

This section focuses on a probabilistic analysis of the Epic Kitchens dataset. The goal is to extract sequences with a high probability of co-occurrence, representing chains of frequently performed actions associated with specific objects. Using a transition matrix, the analysis determines which actions are most likely to co-occur with a certain item and which are most unlikely to be performed. A descriptive statistics analysis is conducted to examine the linguistic aspects that distinguish verbalised actions from those left implicit, to glean more insight from a linguistic perspective.

5.3.1 Extracting Co-occurrences from Epic Kitchens

As the Epic Kitchens dataset contains multiple actions related to daily routine activities, it has been identified only actions related to the core cooking process. Specifically, actions pertaining to utensil cleaning or activities outside direct food preparation, such as *wash dishes* or *put laundry in washing machine*, were removed from the dataset. Following data cleaning, it was possible to compute a transition matrix containing, for each cell M_{i_j}, the probability that the $i-th$ action is followed by the $j-th$ action. High probability (sub-)sequences represent commonly performed action series performed on a certain item. These (sub-)sequences can be taken as a reference to compute which actions, foreseen in a starting plan, may be omitted when generating instructions for another person to follow, as in the case of recipes. Specifically, omitted actions in the instructions sequence are part of a (sub-)sequence for which the transition probability is so high that it is possible to estimate that they

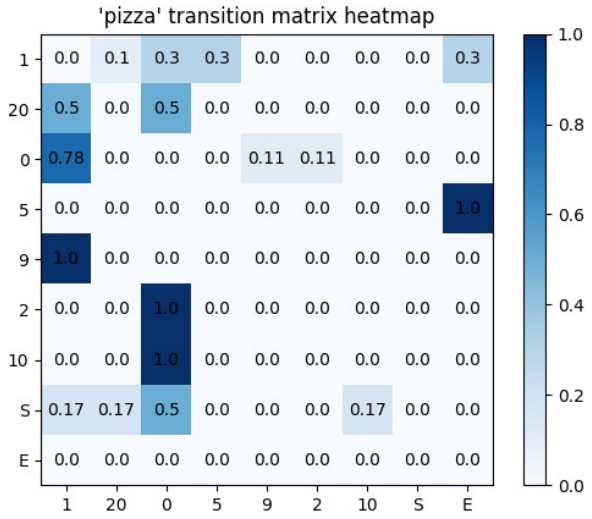

FIGURE 5.9: Transition matrix representing a sequence of states related to the *pizza* noun. Given the size of the dataset, it is reported here only a portion of the matrix for illustrative purposes.

would be performed anyway. Taking into account the matrix shown in Figure 5.9, it represents a transition matrix displayed in the form of a heatmap, depicting the transition probabilities between different states of a process. In this case, it pertains to a sequence of states related to the noun *pizza*. The vertical axis (on the left) represents the start states, while the horizontal axis represents the end states. Each cell in the matrix indicates the probability of moving from the start state (vertical axis) to the end state (horizontal axis). The *S* (Start) represents the initial state of the process while the *E* (End) represents the final or terminal state. The numerical values within the cells indicate the probability of transition between states, where values close to 1 indicate highly probabilistic transitions, while values close to 0 indicate low probability or absent transitions. For example, in cell (1, 9), representing the verb classes *put* and *move*, there is a value of 1.0, indicating a high probability of transition from the start state to the specific end state. In other words, from state 1 (put), there is a 100% possibility that the next action will be 9 (move). In contrast, the value 0.0 in cell (1, E) indicates a transaction to the final state with a low probability. It is therefore 0% possibility to have a sequence of actions that starts with 1 (put) and ends.

Considering the full set of data, it is possible to rebuild the sequence structure of the actions along with their probability of being performed on a specific noun.. To pursue this objective, the Markov chain is employed as it is particularly valuable for modelling stochastic processes across various fields, including linguistics. A Markov chain is a mathematical system consisting of transitions between states

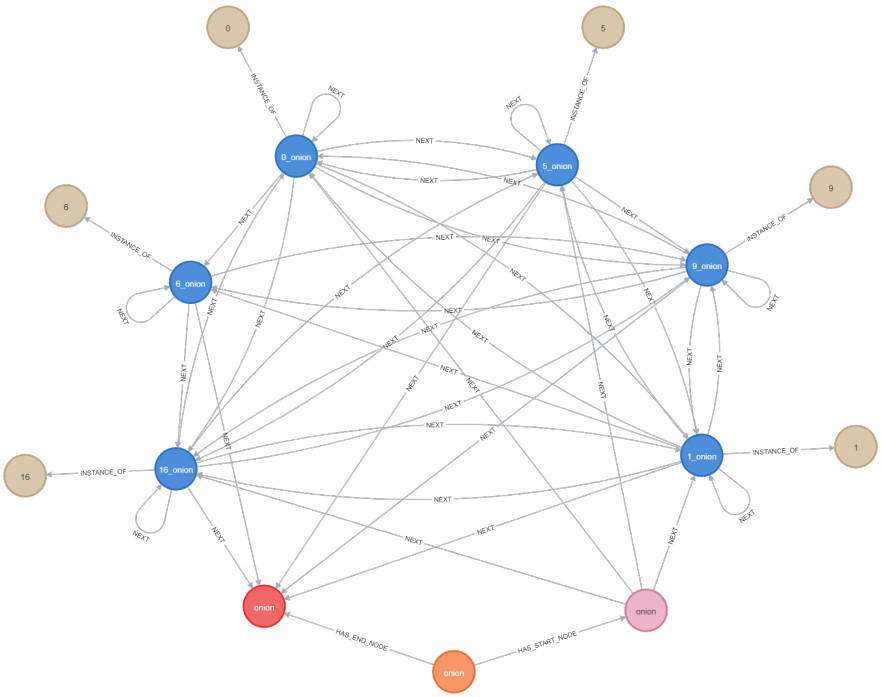

FIGURE 5.10: Graph Representation of the sequences of actions performed on the noun *onion*. Noun_MC (orange) represents a noun (e.g.*onion*), which is connected to MC_Start (pink) and MC_End (red) node through HAS_START_NODE and HAS_END_NODE relationships. These are, in turn, connected to VERB_CLASS_INSTANCE (blue) nodes through the NEXT relationships nodes. Each VERB_CLASS_INSTANCE is connected to the correspondent VERB_CLASS nodes (brown) through the INSTANCE_OF relationship.

within a finite or countable set of states. It is defined by the Markov property, which states that the future state depends solely on the current state, independent of the sequence of events that preceded it. In each Markov chain, the nodes *S* and *E* represent the initial and final states of the process, respectively. For example, a transition such as S $\rightarrow$ 1 can be described as having a probability of 1.0, indicating that the first action linked to a specific noun class has a high probability of involving specific verb classes. Therefore, a Python script was implemented to estimate the transition of actions performed on each noun, as it is a standard approach in data analysis. Markov chains were uploaded into the database, where each node represents a state of the chain and each relationship is a transition of the chain, with the transition probability as a property. An example of action sequences performed on the noun *onion* is illustrated in Figure 5.10. In this graph, the node *Noun_MC* (where MC states for *Markov Chain*) represents a noun linked to MC_Start and MC_End node through HAS_START_NODE and HAS_END_NODE relationships, representing the start/end point of an action for a specific noun. These are, in turn, connected to VERB_CLASS_INSTANCE nodes, which represent each verb class

TABLE 5.3: Nodes and relationships contained in the final database.

Label	Properties
Recipe1M+	Id, primaryName
INGREDIENT	id, primaryName
INSTRUCTION	recipeID, text
FLAVORDB	aliasURL, aliases, flavorDBID, naturalSource, sourceURL
MOLECULE	commonName, complexity, foodDBID, pubchemID, smile
Noun_MC	noun
MC_start	mc_id
MC_end	mc_id
Verb_Class_Instance	verb_class_instance
Verb_Class	verb_class
EPIC_KITCHEN	-
INSTRUCTION	recipeID, id, narration
ACTION	recipeID, id, narration
NOUN	noun
VERB	verb
HAS_INGREDIENT	-
HAS_INSTRUCTION	-
COMPOSED_BY	-
SAME_AS	-
INSTANCE_OF	-
HAS_START_NODE	mc_id
HAS_END_NODE	mc_id
NEXT	mc_id, probability
FOLLOWED_BY	-
HAS_NOUN	-
HAS_VERB	-

contained in Epic Kitchens, through the NEXT relationships. These relationships contain *probability* as property, indicating the probable successor state score given a current state. Each VERB_CLASS_INSTANCE node represents an instance of a verb class, which is connected to the correspondent VERB_CLASS nodes through the INSTANCE_OF relationship. Table 5.3 summarises the nodes and relationship properties contained in the final database.

The information retrieved from the analysis conducted in Section 5.2, along with the Markov chains, are crucial for conducting a preliminary linguistic analysis on the frequency of verb classes in Epic Kitchens, taking RQ3 as a starting point for conducting the investigation.

5.3.2 Investigating Explicit and Implicit Linguistic Features

This section investigates the linguistic characteristics of Epic Kitchens' verb classes. In particular, the attention is on verb classes labelled as INSTRUCTION (i.e., a set of actions that are typically mentioned in a cooking recipe instruction) to identify linguistic features that distinguish them from the verb classes labelled as ACTIONS (i.e., a set of actions that are typically omitted from a cooking recipe instruction). To achieve this, a percentage was calculated based on the individual counts of each

label within each class, allowing identification of the relative weight of each label to the total occurrences for a specific verb class, highlighting which pairs are most representative or dominant.

5.3.3 Methodology

In the Epic Kitchens dataset, each verb and noun is categorised respectively as *noun* and *verb classes*, and each of them is assigned a number, with a total of 124 verbs and 351 noun classes. Out of a total of 124 verb classes, 113 were labelled as ACTION, while 77 were labelled as INSTRUCTION. It is important to highlight that 42 verb classes present only one label: 36 verb classes present the label ACTION, while 6 present the label INSTRUCTION. For this investigation, only the 81 verb classes that present both labels were taken into account. Percentages were calculated based on the individual counts of each label within each class. This calculation makes it possible to identify the relative weight of each label (e.g., ACTION or INSTRUCTION) to the total occurrences for that verb class, highlighting which pairs are most representative or dominant. A high value indicates that a label dominates significantly within the class, while a low value implies less relative importance. This allows for a deeper investigation of the relationships between verbs and contexts of use. Verb classes with the ACTION label are predominant in that they appear most frequently on the entire dataset (more than 95%). In contrast, a smaller percentage (the maximum comes to 67%) is present for verbal classes labelled as INSTRUCTION. Therefore, verb classes labelled as ACTION with a percentage greater than or equal to 85% are considered. As for the verb classes labelled as INSTRUCTION, since they are significantly fewer in number, only the percentages greater than or equal to 56% were taken into account. The data are shown in Figures 5.11 and 5.12.

Verb classes 25 (*add*), 30 (*sprinkle*), 36 (*spread*), 47 (*fry*), 48 (*drain*), 57 (*measure*) and 65 (*turn-down*) labelled as INSTRUCTION, exhibit a notably high percentage compared to those labelled as ACTION. This aligns with the findings from the semantic distribution analysis of actions discussed earlier (Paragraph 5.1.3), where it becomes evident that these verb classes (except for class 57 which is not found in annotated dialogues from the CookDial corpus), are consistently and explicitly expressed in culinary instructions. Focusing on the distribution of ACTION label, it is evident that it appears across a much larger set of verb classes, for a total of 12 verb classes. In particular, the analysis reveals that 6 out of 12 verb classes show

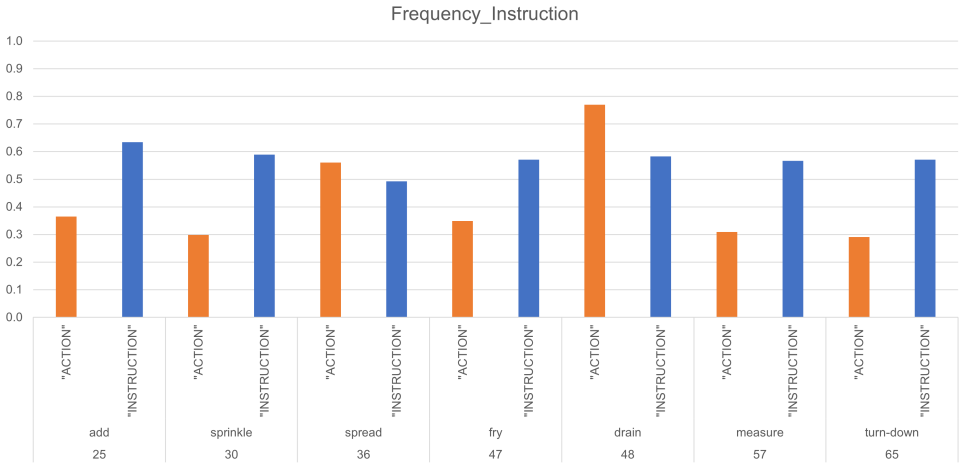

FIGURE 5.11: Verb classes having a frequency greater than or equal to 66% with the label INSTRUCTION. Percentages less than 56% were excluded from the analysis.

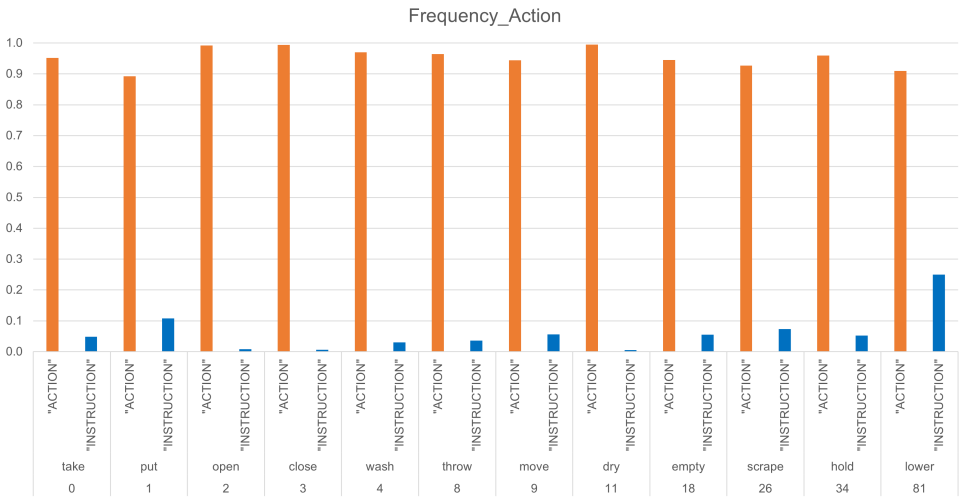

FIGURE 5.12: Verb classes with a frequency greater than or equal to 85% ACTION label. Percentages less than 89% were excluded from the analysis.

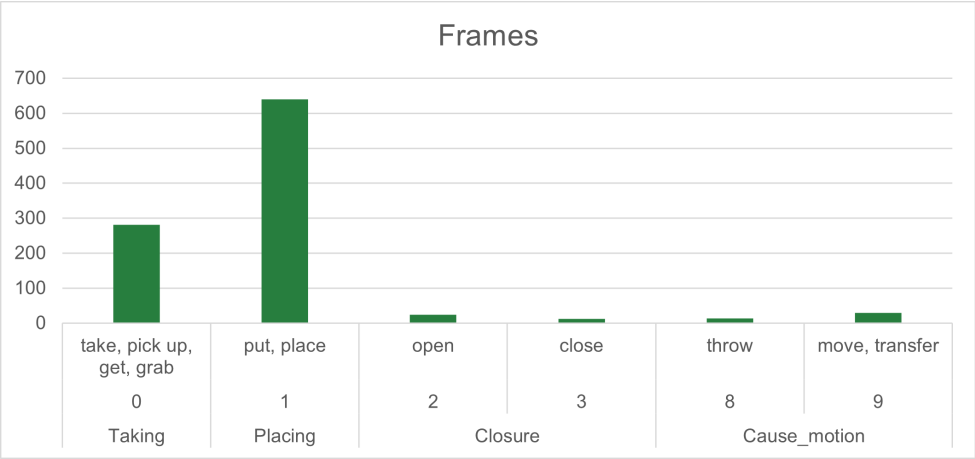

FIGURE 5.13: Occurrences of verb classes along with their corresponding frames.

a relatively small percentage associated with the label INSTRUCTION. The verb classes taken into account are summarized in the box below, providing both the verb class and the corresponding verbs belonging to each class:

Verb classes along with their respective verbs

- **Verb Class 0:** take, pick up, get, grab
- **Verb Class 1:** put, place
- **Verb Class 2:** open
- **Verb Class 3:** close
- **Verb Class 8:** throw
- **Verb Class 9:** move, transfer

Verb classes such as 4 (*wash*), 11 (*dry*), 18 (*empty*), 26 (*scrape*), 34 (*hold*) and 81 (*lower*) are left out in this analysis as they might occur in contexts not strictly relevant to the process of preparing a meal.

The frequency of verb classes labelled as INSTRUCTION across the entire dataset was estimated. FrameNet is employed to provide a structured framework for describing the semantic participants evoked by verbs belonging to each class. The results, illustrated in Figure 5.13, indicate that class 1 (*put*) is the most frequent, followed by class 0 (*take*). Less frequent are class 9 (*move*), class 2 (*open*), class 3 (*close*), and class 7 (*throw*). The verb class 0 evokes the frame *Taking* which depicts

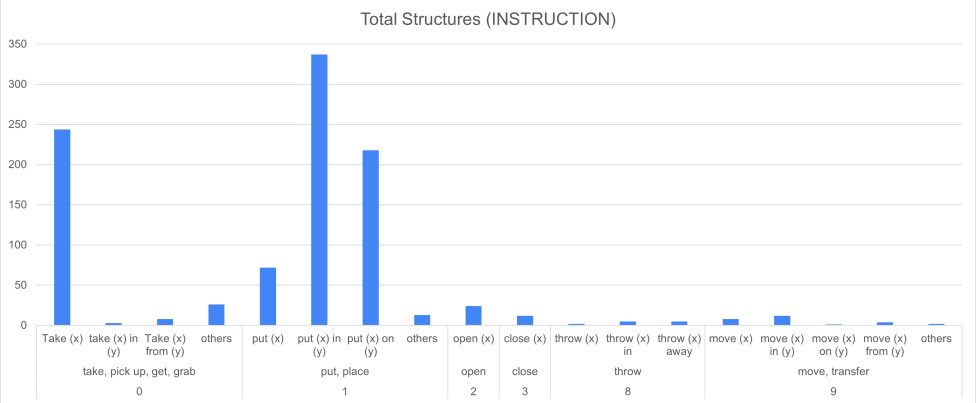

FIGURE 5.14: Total structures occurring with verb classes labelled as INSTRUCTION

a scene where an Agent removes a Theme from a Source with the ultimate aim of taking possession of it. On the other hand, class 1 evokes the *Placing* frame, where an Agent places a Theme at a location (Goal) which is profiled. Classes 2 and 3 evoke the *Closure* frame, where an Agent manipulates a Fastener to open or close a Containing_object (e.g. an oven). Lastly, classes 8 and 9 evoke the *Cause_motion* frame, where an Agent causes a Theme to move from a Source to a Goal along a Path. To determine whether linguistic features differentiate these verb classes from those labelled as ACTION, the study examined the structures in which these classes frequently occur and analyzed their semantic roles.

The syntactic structures for each verb class were identified and estimated their occurrences in the dataset. This approach provides insights into which structures occur and how frequently they appear within each verb class across the entire dataset. The results are summarized in Figure 5.14. The variables x and y correspond to participants playing different semantic roles depending on the frame that the verb evokes. Structures consisting of one participant include only the variable x (e.g., take *the flour*). In contrast, structures involving two participants introduce the variable y (e.g. put the flour *in bowl*). Structures labelled as *others* represent diverse structures that occur with limited recurrence in the dataset and were therefore excluded from this investigation.

5.3.4 Results

The analysis first takes into account structures involving one participant, which the results are illustrated in Figure 5.15, displaying the quantitative distribution across

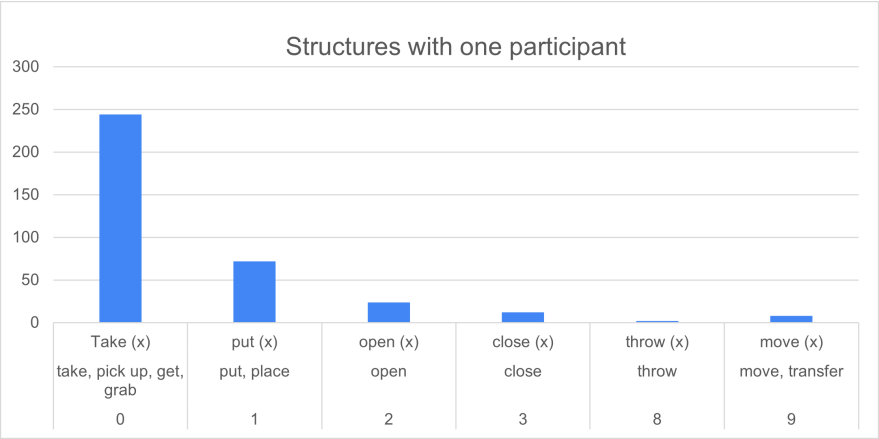

FIGURE 5.15: Structures with one participant

related verb classes. The data highlights a predominance for classes 0 and 1. In contrast, less frequent occurrences are associated with other verb classes, such as classes 2, 3, 8 and 9. For this reason, the focus is on verb classes 0 and 1.

The roles played by the variable *x* are THEME for classes 0 and 1 as the LU evoked by verbs are respectively *Taking* and *Placing*. Analyzing the Epic Kitchens corpus reveals that, for verb class 0, the THEME is typically omitted in the subsequent action. For instance, in the action chain *take onions - transfer in bowl*, the THEME (*onion*) explicitly stated in the previous action is omitted in the following one. Conversely, for verb class 1, the GOAL in which the THEME has to be positioned is expressed in the preceding action. For example, in the action chain *open oven - put tray*, the GOAL (*oven*) is mentioned earlier. Therefore, explicitly mentioning verbs belonging to these verb classes completes the meaning of the preceding or following actions. Analysing the verb classes labelled as ACTION, the frequency with which this structure occurs is significantly higher than previously observed, particularly for verb class 0, as illustrated in Figure 5.16. Although the semantic frames are the same for ACTION labels, the relationship between previous and subsequent actions, is absent for this label, as the participants do not appear to be related in any way to either previous or subsequent contexts, highlighting the different functions between the labels.

Structures showing a preposition *in* are present with verb classes 0, 1, 8, and 9, as shown in Figure 5.17. Preposition *on* appears in association with classes 1 and 9 (Figure 5.18), while *from* is limited to classes 0 and 9 (Figure 5.20). Preposition *in* predominantly occurs with verb class 1, while its use with other verb classes is

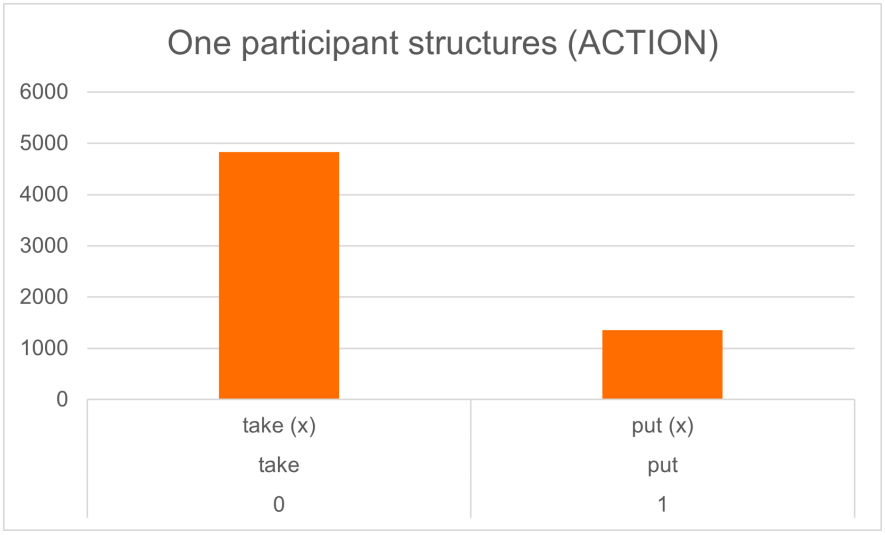

FIGURE 5.16: One participant structures ACTION

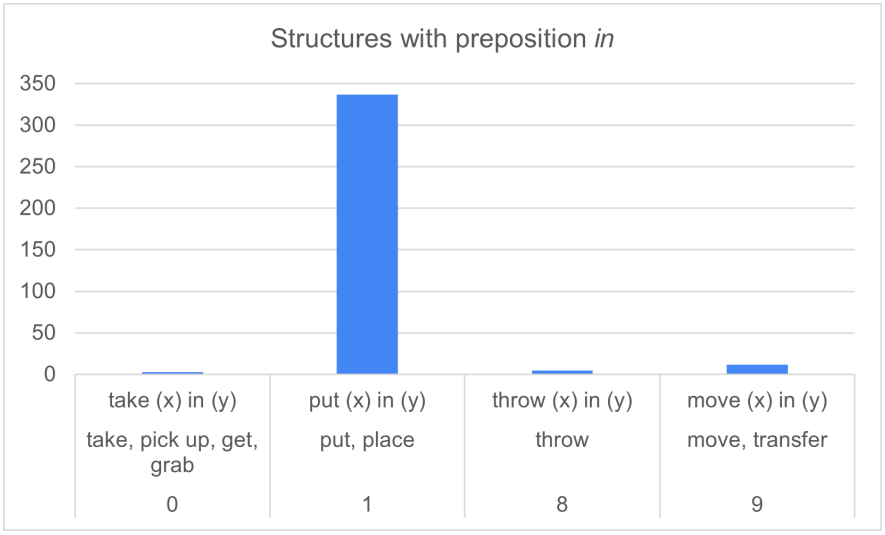

FIGURE 5.17: Structures with preposition *in*

minimal. A similar pattern is observed with preposition *on*, where verb class 1 dominates, leaving verb class 9 significantly less represented. Preposition *from* is absent in constructions involving verb class 1 and it is primarily associated with verb class 0, which appears more frequently than verb class 9. However, the overall frequency of this preposition is relatively low, with fewer than 10 occurrences. Given that this structure occurs predominantly with verb class 1, it is considered appropriate to focus on this class, as it emerges as the most recurrent. Which concerns structures showing the preposition *on*, several structures emerge for verb class 1, as shown in Figure 5.20. The analysis reveals that the most frequent structure involves explicit specification of the participant involved. By contrast, structures with zero anaphoric referents (e.g., put ϕ on stove) or pronominal referents (e.g., put *it* on stove) occur significantly less

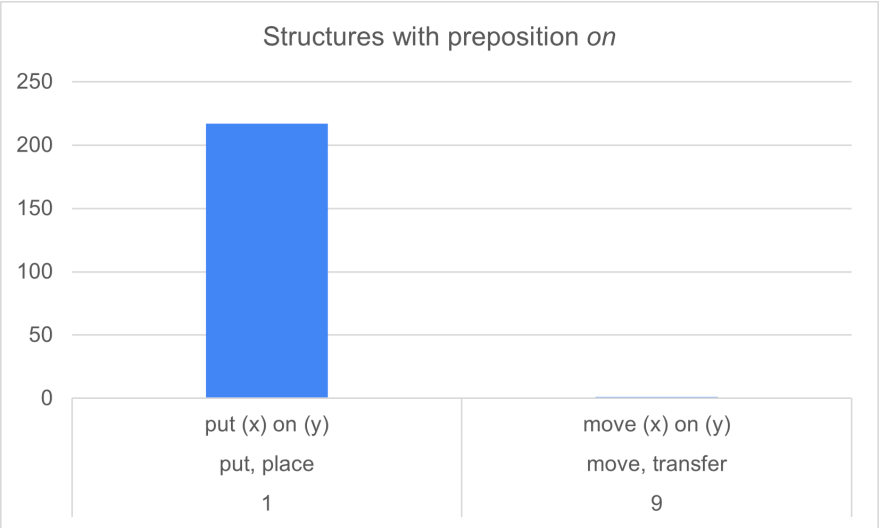

FIGURE 5.18: Structures with preposition *on*

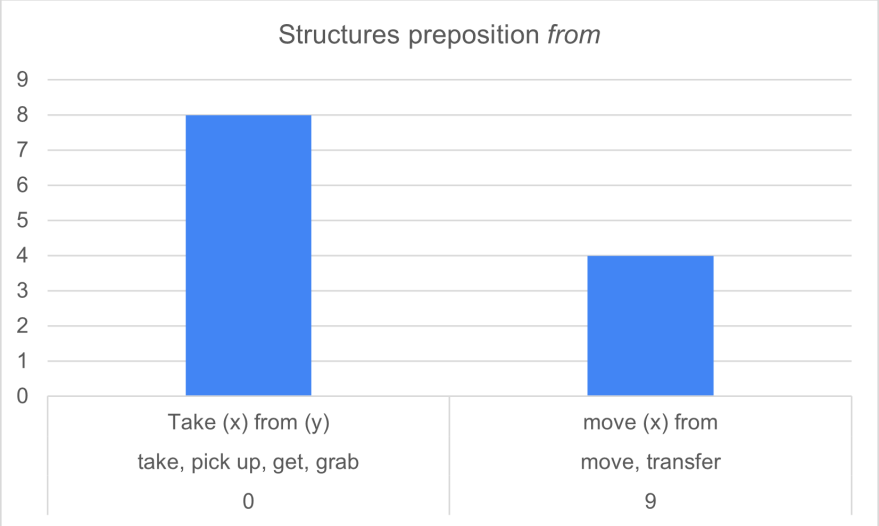

FIGURE 5.19: Structures with preposition *from*

often. Similarly, structures where the position of the participant is specified (represented as *[position]*), are less commonly used (e.g. put the cherry syrup *on top of* the pancake). These structures confirm the presence of the structures seen above, where the participant THEME is made explicit in a sequence of actions immediately preceding those in which the THEME is absent (zero anaphora) or through the use of a pronominal referent.

Taking into account verb class structures labelled as ACTION, out of a total of 5.308 units, the occurrences were estimated and are reported in Figure 5.21. Although the occurrences of this label are more frequent than those of the INSTRUCTION label, the structure where only one participant is needed is preferred to structures

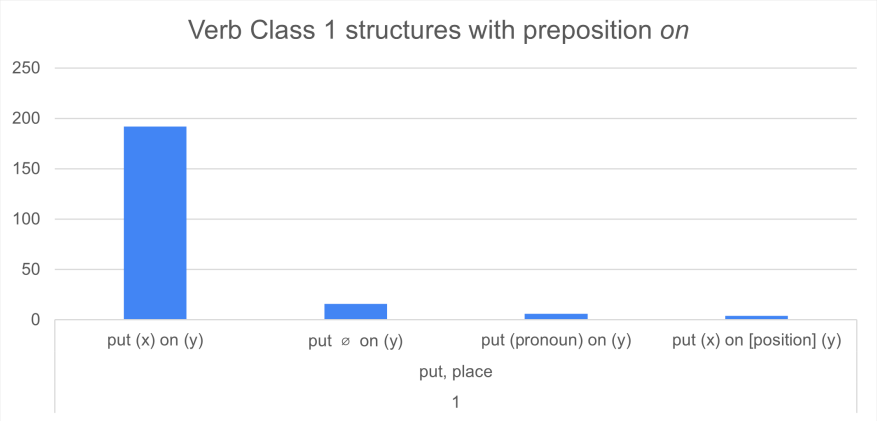

FIGURE 5.20: Verb class 1 presenting preposition *on*

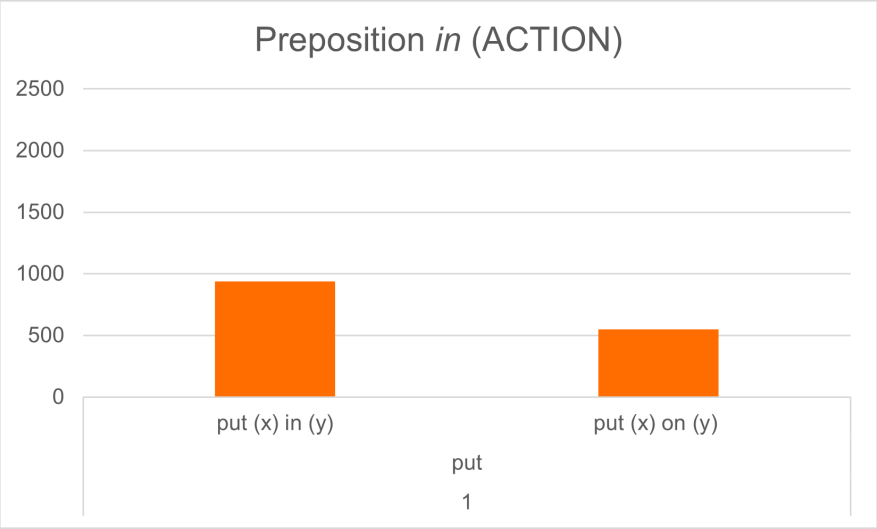

FIGURE 5.21: Verb class 1 structures labelled as ACTION

with two participants with the prepositions *in* and *on*. The linguistic features of verb class 1 labelled ACTION show no difference from the features found for the one labelled INSTRUCTION. In this regard, they evoke the same *Placing* frame. Based on these results, the distinction between the two labels can be summarised as follows:

1. Verb class 0 is labelled as INSTRUCTION when the participant of the subsequent action is omitted and cannot be automatically inferred from the linguistic context in which verbs belonging to different verb classes occur. Conversely, it is labelled as ACTION when the action can be inferred as it is unrelated to the immediately following action;

2. Verb class 1 is labelled as INSTRUCTION when the goal is not explicitly stated in the immediately preceding action and cannot be automatically inferred from

the linguistic context in which the action of verb class 1 occurs. Conversely, it is labelled as ACTION when the action is unrelated to the preceding context.

5.3.5 Discussion

As the Epic-Kitchens dataset categorises verbs and nouns as *noun* and *verb classes*, which of them is assigned a number, a Python script was implemented to extract a co-occurrence matrix of verbs along with nouns and integrated with the other sources in the database, employing Markov chains. Based on the results obtained in Section 5.2, a probabilistic estimation of the actions has been derived, making it possible to identify the co-occurrences of actions with a given noun. This made it possible to delve deep into the data, detecting the linguistic aspects that distinguish verbalised actions from those left implicit (RQ3). In this respect, a descriptive statistics analysis of the Epic Kitchens data has been conducted, with a particular focus on verb classes 0 and 1.

Results suggest that verb class 0 is labelled as INSTRUCTION when the THEME of the subsequent action is omitted and cannot be automatically inferred from the linguistic context in which verbs belonging to different verb classes occur. Conversely, it is labelled as ACTION when the action can be inferred as it is unrelated to the immediately following action. Verb class 1 is labelled as INSTRUCTION when the GOAL is not explicitly stated in the immediately preceding action and cannot be automatically inferred from the linguistic context in which the action of verb class 1 occurs, as happened with verb class 0. On the other hand, verb class 0 is labelled as ACTION when the action is unrelated to the preceding context. For instance, in the action chain *take onions - put in bowl*, the THEME (*onion*) and GOAL (*bowl*) are explicitly stated in the previous/next action. In a communicative interaction, this information is omitted because the information can be easily retrievable from the previous/next action if and only if both sentences are made explicit. For this reason, they are labelled as INSTRUCTION. On the other hand, in the action chain, such as *put knife - take colander*, the two actions are not directly connected. The context in which these actions co-occur does not affect the understanding of the sentence's meaning. Therefore, these actions are labelled as ACTION as they are irrelevant for this purpose.

These results align perfectly with the previous analyses described in Sections 5.1 and Section 5.2. We observe that the actions follow a logical order, as evident in the

distribution of semantic frames within the dialogue flow (Section 5.1). Even though the actions in Epic Kitchen do not refer to any precise instructions and thus lack a pre-established order based on recipe's instructions (as happens with CookDial), a logical flow can be deduced. Taking into account the actions chain such as *take knife - cut onion*, the action *take the knife* is labelled as ACTION, while *cut onions* is labelled as INSTRUCTION, as it contains core information. More specifically, in an action planning, the action of cutting an object must be preceded by an action that enables the taking of an instrument to carry out the action. Knowing if an object can be cut (Section 5.2) as well as the knowledge that an object cannot be cut without a sharp tool capable of cutting it - and, therefore, following a logic flow, such as (i) taking the tool and (ii) cutting the object -, is already shared in the speaker's mind, constituting what has been described as *Presupposed Knowledge*. In this regard, thanks to a probabilistic estimation, it was possible to assess the probability of the co-occurrence of actions on an object. Through a linguistic approach, it was possible to describe the linguistic particularities that highlight the differences between implicit and explicit information, answering RQ3. Observing the syntactic structures of the sentences along with their semantic roles made it evident that certain structures must inevitably be made explicit to understand the task at hand, while others can be omitted. This last analysis confirms that many actions labelled as ACTION are, in reality, incorporated in the evoked semantic frame (in this case, *Cutting*) and, thanks to the individual analysis, allows us to state that it is irrelevant to make them explicit to understand the task to be performed.

CHAPTER

SIX

TOWARDS THE IMPLEMENTATION OF CONVERSATIONAL AGENTS

The analysis presented so far has laid the groundwork for implementing a conversational agent. Optimising the identification of CS instructions that can be omitted to maximize utility can be achieved within a dialogue system. As an example application, it has been taken into account Bastian, an embodied conversational agent developed within the FANTASIA plugin, specifically designed for solving tasks in the cooking domain.

6.1 FANTASIA architecture

FANTASIA (Framework for Advanced Natural Tools and Applications with Social Interactive Agents) [227, 230] is a plugin for the Unreal Engine[1] designed to support the development of Embodied Conversational Agents (freely available on Github[2]). The Unreal Engine is a platform for creating immersive Real-Time Interactive 3D (RTI3D) experiences. The introduction of Metahumans by Epic Games has further enhanced the capabilities of Virtual Humans to serve as conversational interfaces. As illustrated in Figure 6.1, FANTASIA aims to combine various components, including a graph database, a dialogue manager, a game engine, and a voice synthesis engine, to create social interactive systems. While most approaches currently rely on LLMs, FANTASIA enables the integration of multiple AI tools into the Unreal Engine real-time infrastructure. Among other features, FANTASIA provides a connector that

[1] www.unrealengine.com/en-US/
[2] github.com/antori82/FANTASIA

allows for Neo4j graph database queries from within the Unreal Engine, as well as functionality to dynamically assemble and query Bayesian networks. FANTASIA has been used in several projects, including VISIT3D (Virtual Interactive System for the Implementation of Tours in 3D) [3] owned by Logogramma s.r.l [4], where the Virtual Reality experience is enriched by the presence of a virtual agent – a meta-human – who accompanies the visitor in their exploration of the works, responding to their evolving requests through the communicative interaction.

The key principles followed by a conversational AI built with FANTASIA are:

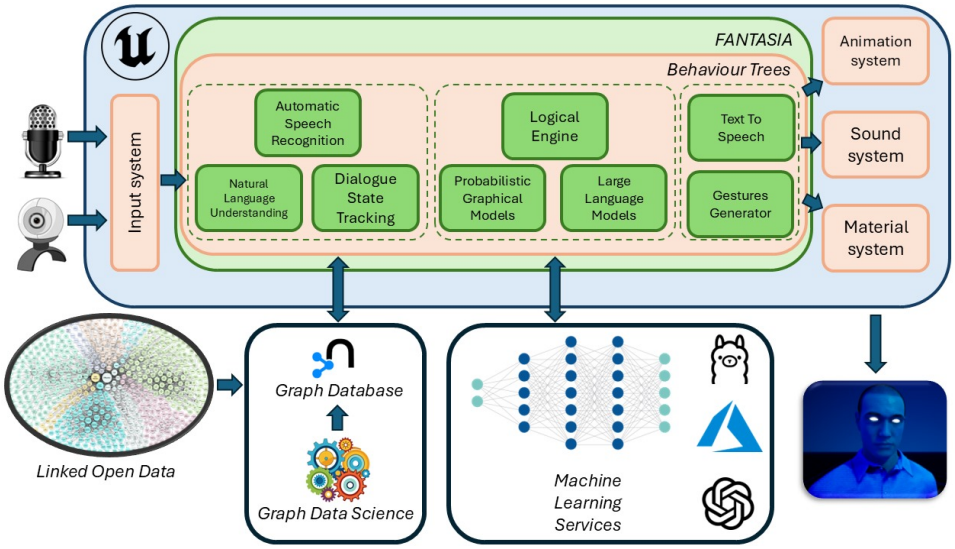

FIGURE 6.1: The FANTASIA architecture

- **Behaviour Trees (BT)** [103] are employed to organise and prioritise subtasks. This concerns hierarchically structuring the sequence of checks needed to generate clarification requests, i.e. when a speaker did not (fully) understand or is uncertain about what was previously said or meant with an utterance [108].

- **Graph Databases** are employed for knowledge representation and dialogue state tracking. By integrating the structure of the knowledge domain with the way users reference it, relevant subsets of the available information can be extracted. This enables reasoning over those subsets to generate the system's next utterance.

[3] Visit3D: https://www.visit3d.it/
[4] Logogramma: https://www.logogramma.com/

- **Bayesian Networks (BN)**, implemented with the aGRuM library [82], serve as decision models to determine the most *useful* system action based on the desired goals. These networks, along with their variants such as Influence Diagrams, can be dynamically constructed from knowledge sub-graph structures.
- **Large Language Models (LLMs)** are employed to articulate the decisions made by probabilistic graphical models.

6.1.1 Interaction Model in FANTASIA

The FANTASIA interaction model manages the dialogue as a physical system in which the participants exert forces on a network structure, pushing it through information exchanges towards an equilibrium state that exhibits the desired goal pattern [226]. Participants use dialogical moves to intervene in this graph and guide it towards a desirable, stable state. The graph configurations are then checked following a linguistically motivated priority order to determine the best utterance to produce to reach the desired configuration. The different types of graph configurations can be summarized as follows [41][p.86]:

- **Interpretability**: a graph is uninterpretable if any of the graph patterns describing each of the foreseen communication problems is activated. In these cases, a Clarification Request (CR) is produced;
- **Completeness**: an interpretable graph is incomplete if information needed to respond to the user intent is missing. In these cases, a request for information is produced;
- **Coherence**: a complete graph is incoherent if logical conflicts are found in the *belief graph*. In these cases, the adequate disambiguation question is produced;
- **Stability**: a coherent graph is unstable if there are open issues, like unanswered questions. In these cases, a question answering strategy is activated;
- **Desirability**: a stable graph is undesirable if it does not exhibit the goal pattern. In these cases, the most *useful* dialogue move to create the goal pattern is produced, like an exploration or exploitation move.

From Interpretability to Desirability, the machine's motivations for continuing the dialogue are increasingly driven by self-oriented objectives. Solving communicative problems implies a lower degree of illocutionary force compared to taking

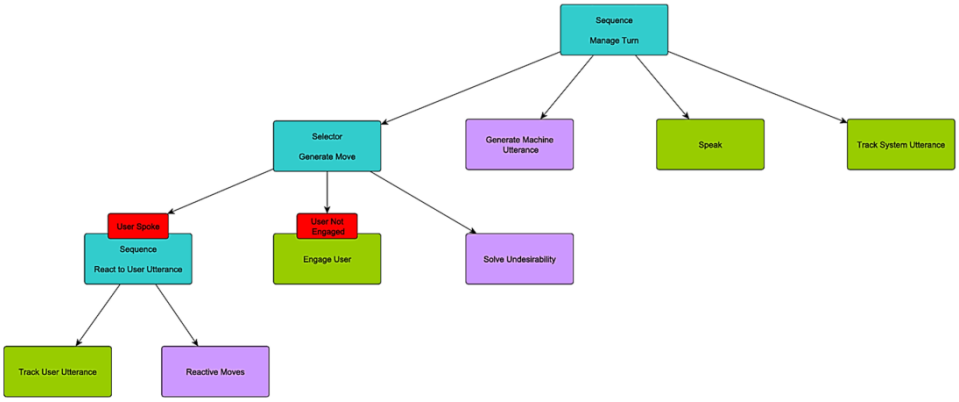

FIGURE 6.2: The Behavior Tree handles the machine turn. If the turn was taken because the user spoke, the Natural Language Understanding interpretation is tracked in the Graph Database, and the subtree for Reactive Moves takes priority. If this fails or if the user does not speak, the system considers the necessity of engaging the user in conversation. If this also fails, the BT for solving Undesirability is activated. Once a machine action is selected, the BT generates the actual utterance, which the system then speaks and tracks in the graph database. (Source: Di Bratto, 2024 [41].)

a stand and answering questions[5]. In this respect, task hierarchies in FANTASIA can be efficiently modelled through Behaviour Trees. The management of dialogue turns is grounded in the turn-taking system theory [266], a fundamental framework for understanding conversation dynamics. When it is the system's turn, it assesses whether a response to the user's statement is necessary, then records its interpretation in the graph database along with the recognised intents and entities. Subsequently, the system assesses whether the user's move requires a specific type of response, relating to the configurations from non-interpretability to instability. If the system is not waiting for any input from the user, it checks whether taking the turn involves the user, and acts accordingly. Otherwise, the system activates strategies to resolve an unwanted graph configuration, as shown in Figure 6.2.

To address the issue of non-interpretability, a hierarchical classification of communicative functions such as Clarification Request (CR) is adopted, as described in Di Maro [73]. This hierarchy corresponds to the communicative levels of contact, perception, comprehension, and intention. Each of these hierarchies can present one or more problems, which may be caused by specific linguistic issues or relate to missing information, e.g. lexical comprehension, reconstruction of references, syntactic comprehension and logical comprehension. When non-interpretable graph configurations are present, specific patterns are verified and subsequently mapped to speech

[5]More detailed information on this subject can be found in Di Bratto, 2024 [41].

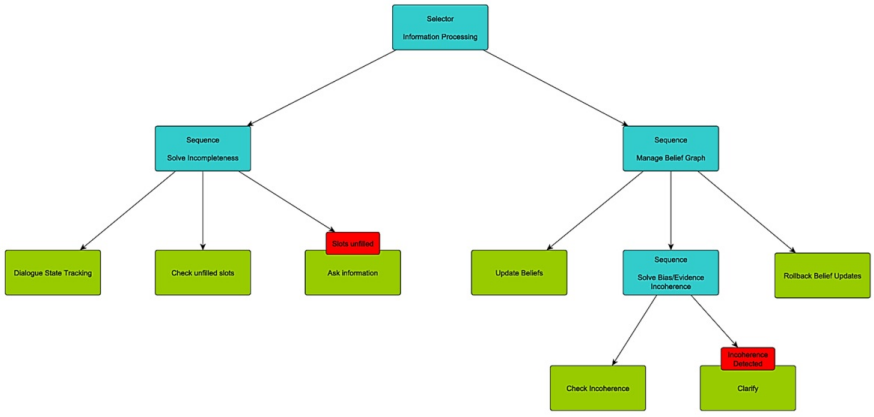

FIGURE 6.3: The BT for Reactive Moves first checks for interpretability problems and generates a Clarification Request following the priorities described in Di Maro [73]. The check/clarification pattern is simplified in the Figure for readability. A dedicated subtree handles Information Processing problems. If there are no interpretability issues, the subtree handling Instability is activated. (Source: Di Bratto, 2024 [41].)

acts, also codified in the Behaviour Tree, as shown in Figure 6.3.

The first type of Information Processing problem is *incompleteness*, which is addressed by activating Dialogue State Tracking strategies. The apparent lack of crucial information in a user's statement can occur as this information is already present in the context [41]. If an object has already been mentioned, it is unnecessary to explicitly state the object in question as the subject of each subsequent statement. For instance, if a pan was already mentioned, there is no need to mention it again when advising to add ingredients. Rather, as emerged from the analysis conducted in Chapter 5.3, it is preferred to directly indicate the next action to be applied to the previously mentioned object, e.g. oil to grease it. Information can be recovered by making queries that traverse the graph, as the FANTASIA interaction model maintains this information in the nodes of the graph database. Specifically, the nodes in the Neo4j graph, designed to represent elements, have a property containing a Cypher query that executes specific strategies for that element, ensuring Dialogue State Tracking. From the interaction model perspective, the task involves identifying non-full nodes and executing corresponding queries that are not domain-dependent. Consequently, if there are still non-full nodes in the graph, a clarification request is generated [41, 74]. Otherwise, the strategy that manages the inconsistency is activated, as shown in Figure 6.4.

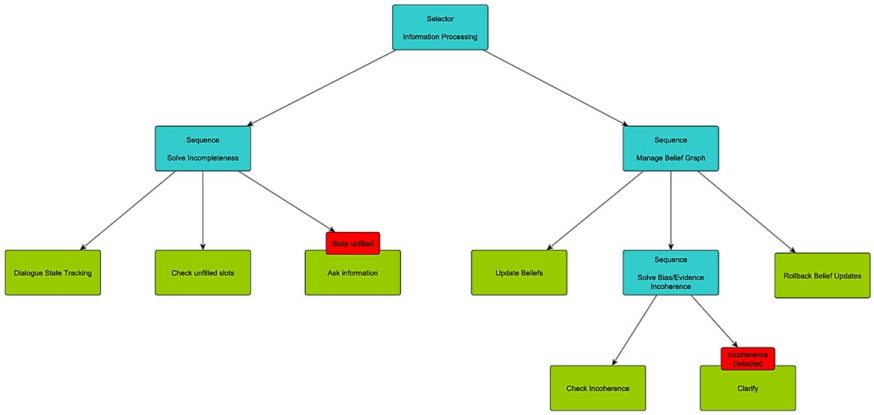

FIGURE 6.4: The BT for Information Processing problems involves the following steps: First, any incompleteness in the information is checked and, if necessary, resolved by attempting to apply Dialogue State Tracking or by generating an information request. If the graph is complete, the belief graph is updated coherently with the user utterance. Then, the belief graph is checked for any incoherence. If an incoherence is found, a clarification request is generated, and the belief graph updates are rolled back. (Source: Di Bratto, 2024 [41].)

When there are no comprehension issues, the graph is checked for any *instability*, primarily due to the need to address the questions asked by the user. From a linguistic perspective, the principles of Grice's Conversational Maxims represent a fundamental norm for achieving cooperative and effective communication, as discussed in Chapter 2. In this configuration, the answers to the questions are obtained by accessing structured data or using Retrieval Augmented Generation to extract information from unstructured data, as shown in Figure 6.5. This marks the first instance where LLMs are employed in the FANTASIA interaction model. As already mentioned in Chapter 1, research has demonstrated the great potential of Q&A using LLMs, although their typical weakness of producing hallucinations also affects FANTASIA applications. Notably, studies have emphasized the importance of supporting the generative process using LLMs by incorporating information from structured data sources. The basic idea, which is still in the experimental phase, is to extract knowledge from a knowledge graph and use it, through an automatically generated prompt, to constrain the LLM's response to the provided dataset, typically in document form. Furthermore, FANTASIA places RAG approaches in the instability subtree, treating them as a fallback strategy to reduce the possibility of errors in case answers are not found in the database (e.g. names, dates, etc.), rather than as a dialogue management technique. Extraction of documents relevant to the user's query is performed by leveraging similarity search through vector indexes. This technique is widely used in

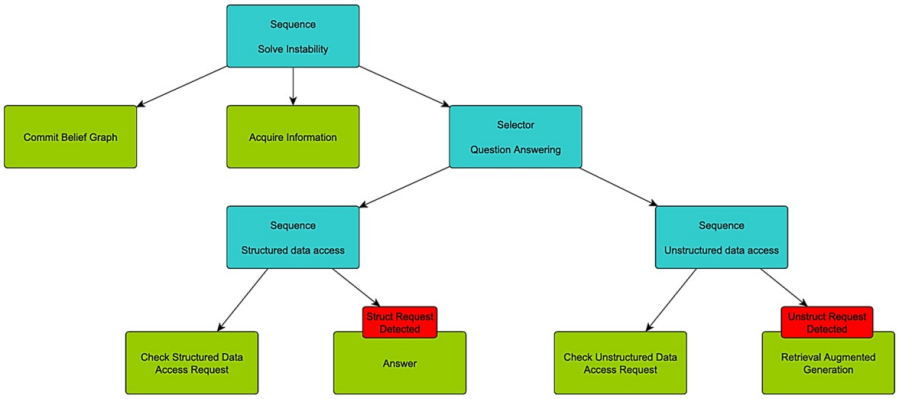

FIGURE 6.5: The BT to manage Instability. If the graph is coherent, changes to the belief graph are committed and information contained in the user utterance is saved in the graph, if necessary. The system then checks if the user asked a question, and in this case, it activates either the strategy dedicated to catalographic data extraction or the RAG strategy. (Source: Di Bratto, 2024 [41].)

RAG applications and can be generalized to the more abstract problem of extracting relevant subgraphs.

If no open issues are detected, the system can produce a move to direct the dialogue towards the desired goal (i.e. *Desirability*). For example, it can have the user accept a proposed stance, such as a recommendation, or it can ask a question when the available information is not sufficient for the system to take a stance. Deciding which move to perform, when communication problems are not present, depends on the goals of the system when producing linguistic moves and involves activating a strategic decision mechanism [41]. In other words, deciding which move to execute when there are no communication problems depends on the purpose behind the system's *linguistic moves* [41]. Actions aimed at altering the representation of the context in the graph, to maximize the likelihood of desirable patterns being displayed, differ fundamentally from ML reactions and support a hybrid vision of Conversational AI. In this regard, making decisions involves computational models designed for decision-making and for combining structured knowledge with real-world observations subject to noise, also originating from ML. Consequently, probabilistic graphical models – such as Bayesian networks and Influence Diagrams – are employed to implement goal-oriented decisions for dialogue management. Unlike Neural Networks, PGMs can be assembled at runtime and populated with prior probabilities and evidence based on the situation at hand. Figure 6.6 shows how Undesirability is represented in terms of BT.

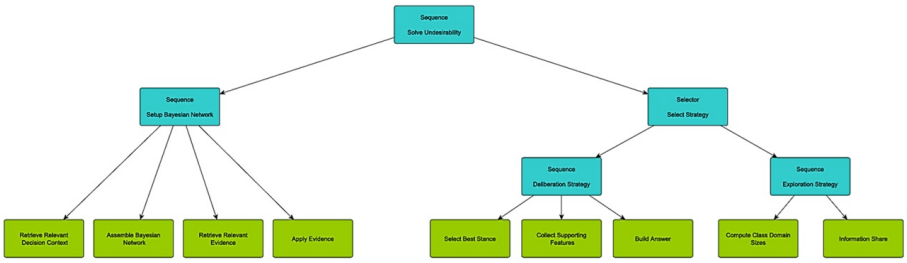

FIGURE 6.6: The BT to manage Undesirability. The relevant decision context, a subgraph of the whole knowledge domain, is extracted from the graph database using the collected beliefs and given the target goals. The structure of the subgraph is used to dynamically assemble a Probabilistic Graphical Model (e.g., a Bayesian Network) and to apply evidence to it. Given the final configuration of the decision model, the system can either *deliberate* (take a stance and try to concretise the goal pattern), possibly using supporting arguments to sustain its position, or it can explore the domain by asking the most *useful* question to reduce the decision model entropy. (Source: Di Bratto, 2024 [41].)

In this respect, the Behaviour Tree for resolving undesirability has been implemented with a set of tasks that update the decision-making strategy [6]. Figure 6.6 illustrates the use of the Setup Bayesian Network sequence, which has been expanded, taking into account progress made in the work conducted by Giannini et al. [114]. A description of the procedure is explained as follows:

1. Prolog generates an action plan for a specific goal and sets the sequence of instructions necessary to build the Influence Diagram;

2. Instructions are executed to assemble the Influence Diagram, whose structure is defined in the planning phase;

3. The Conditional Probability Table of the nodes is populated to represent the situation of interest, and the utilities for the decisions to be made are established;

4. Inference algorithm is applied for the decision making.

In other words, the inference engine can be used at runtime to plan a strategy and assemble probabilistic graphical models, which can then address the practical resolution of unwanted aspects associated with the conversation. An implementation of this approach is shown in Figure 6.7.

[6]The description provided here is a part of the work carried out by Danilo Esposito for his Master's Thesis in Computer Science, supervised by Prof. Antonio Origlia, who deserves credit for this implementation.

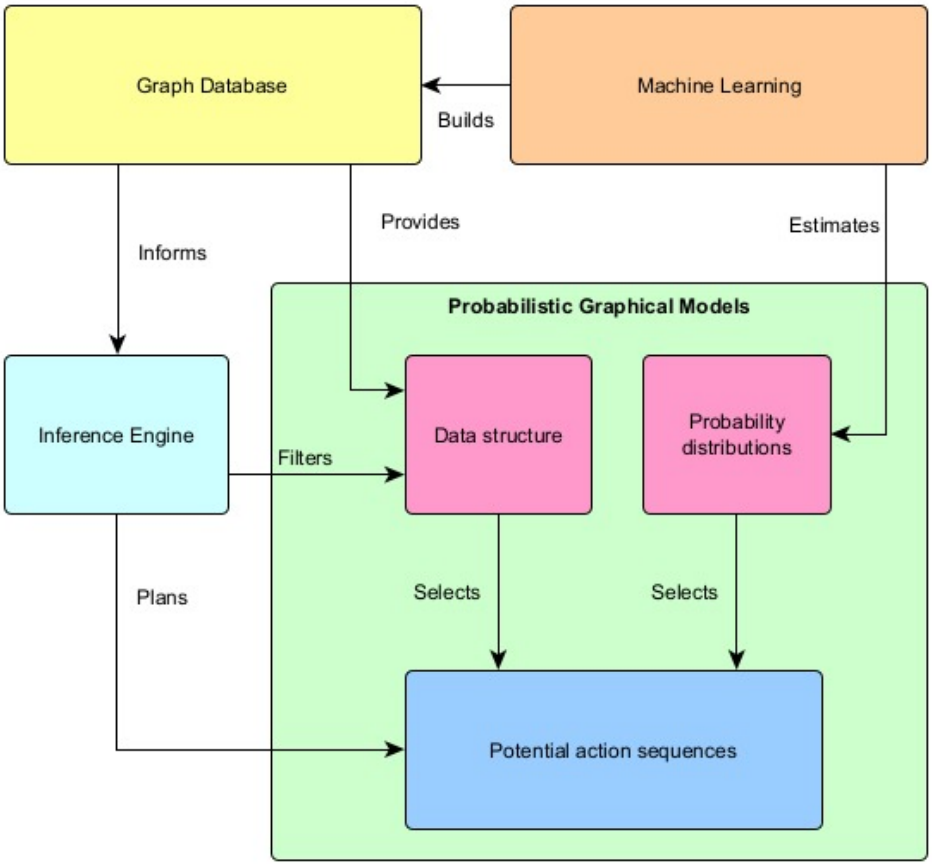

FIGURE 6.7: Diagram describing a decision-making system based on probabilistic models and machine learning.

The Inference Engine generates all possible sequences of actions based on the assumption that specific things are true or false, as discussed in the assumption-based reasoning in Chapter 4. These sequences are then supplied to the Probabilistic Graphical Models. From this perspective, the inferential engine is also - and primarily - used to generate a program that constructs probabilistic graphical models, such as an Influence Diagram. The Graph Database, from the inferential engine's viewpoint, serves as its informant. The planner does not operate on probabilities but rather has certainty about the truth of the information while needing to generalize and remain consistent with the representation of knowledge. Despite the uncertainty introduced when transitioning from the database to the graph, the Probabilistic Graphical Models select the sequences of potential actions, also merging their decisions based on the probability distributions estimated by the Machine Learning model used to enable reasoning under uncertainty.

6.2 Identifying CS Instructions to Follow a Principle of Maximum Utility: The Case of Bastian

As seen in Chapter 1, classical AI was more adept at long-term planning and had the advantage of providing clearly interpretable reasoning to motivate its actions, as well as less imposing requirements in terms of computing power and data availability. In this respect, plan generation is typically managed by logical engines, which effectively implement reasoning mechanisms based on assumptions [246]. These systems generate all potential solutions by considering temporary assumptions regarding the truth value of various propositions. Due to its need for exact definitions for everything, it still failed to provide a viable approach for applications that had to deal with physical reality, which was disputed over the years as the knowledge of the world is impossible to represent efficiently in its entirety [43]. While earlier approaches employed logical engines to directly derive solutions, the probabilistic nature of CS, as outlined here, complicates this approach. It also echoes what was addressed in Chapter 3, namely that the research fields of KR and CSR are partly interconnected, the role of logical engines within the architecture proposed here is to dynamically generate decision models that facilitate reasoning under uncertainty. As an example application of managing kitchen work collaboratively, it is here taken into account Bastian (Figure 6.8), an embodied conversational agent built in FANTASIA, who can plan a sequence of operations to solve a task, but does not have to carry it out alone and instead must instruct someone else. Assuming the complete plan is ready in the agent's control centre, it is necessary to decide which instructions to provide the interlocutor to complete the task with the fewest instructions.

6.2.1 Methodology

As reported in Luger [193], the task of a planner is to determine a sequence of actions that enables a solver to carry out a specific task. Planning is a critical element of AI systems, playing a pivotal role in the reasoning phase and offering a wide range of applications. Taking into account the robotic field, plans are crafted by searching through a vast space of possible actions until the sequence needed to complete the pre-established task is discovered. This expansive search space represents the various states of the world that are altered by applying any available action. The larger the search space, the more complex the problem becomes, and the more crucial it is to

FIGURE 6.8: Bastian, the Embodied Conversational Agent built in FANTASIA.

carefully track the changes that occur when each action is performed. In the following example, the world of the robot is limited to a set of blocks on a table, and the available actions are restricted to the robot's single arm, which can: pick up a block `(pick up (X))`, put down a block `(put down (X))`, and stack a block on top of another `(stack (X,Y))`. Starting from a Simple Prolog Planner[7], these sequences can be explained as follows:

- A set of simple moves with the syntax `move(Name, Preconditions, Actions)`, having a name, a list of preconditions and a list of postconditions;
- Rules for keeping track of changes in state and of the current state, with the management of lists, stacks and sets;
- the main code of the planner, as reported in Listing 6.1.
- recursively executes the planning phase with the updated variables, as reported in Listing 6.2.

[7]Simple Prolog Planner: https://www.cs.unm.edu/ luger/ai-final/code/PROLOG.planner.html [last visited on: 22/03/2025]

```
plan(State, Goal, _, Moves) :-
    equal_set(State, Goal),
    write('moves are'), nl,
    reverse_print_stack(Moves).

plan(State, Goal, Been_list, Moves) :-
    move(Name, Preconditions, Actions),
    conditions_met(Preconditions, State),
    change_state(State, Actions, Child_state),
    not(member_state(Child_state, Been_list)),
    stack(Child_state, Been_list, New_been_list),
    stack(Name, Moves, New_moves),
    plan(Child_state, Goal, New_been_list, New_moves), !.
```

LISTING 6.1: The first rule, which is placed higher in the code, checks if `State` and `Goal` are two equivalent sets, i.e. if the objective state has been reached: if so, it prints the explored `Moves`. The second rule, i.e. the planning phase itself, is a recursive rule that: (i) Explores the moves, (ii) Changes the state if the pre-conditions are met, applying the post-conditions (modifying the `State`).

```
go(S, G) :- plan(S, G, [S], []).
```

LISTING 6.2: `S` is the list representing the initial state and `G` is the list representing the goal state.

The planner shown here, which operates in the context of pure logic programming, bases its functioning on strings. In other words, each element of the state is a string, so it is not possible to exploit unification in Prolog to verify facts in the knowledge base and to make new assumptions at runtime with `assert/1` and `retract/1`, at the basis of assumption-based reasoning.

The planning proposed by Luger [193] has been taken into account as a starting point and extended in this application for developing a planning allowing the agent to infer the information contained in the real-time knowledge base, rather than managing the state elements as strings (Listing 6.3). This planner was called *Bastian Planner* and a result is shown in Listing 6.4 [8]. The `verify(Preconditions)` rule is used to verify if the preconditions of the move, taken into consideration in this iteration, are satisfied based on what has been asserted up to that point in the computation. After changing the state, we continue by moving forward with the recursion for the

[8] This part was also extracted from the work carried out by Danilo Esposito, who implemented the planner reported here.

next move to be selected, performing a `rollback()` in case the recursive step is not successful. Lastly, it is employed a rule (Listing 6.5) to simplify the execution.

```
    %%% init: used to assert each fact in the initial State %%%
init([]).
init([H|T]) :- assert(H), init(T).
%%% verify: used to perform the inference %%%
verify([]).
verify([H|T]) :- H, verify(T).
%%% rollback: used to: %%%
%%% 1. Assert again all retracted facts and
%%% 2. retract again all asserted facts in the recursion %%%
rollback([]) :- false.
rollback([add(P)|T]) :- retract(P), rollback(T).
rollback([del(P)|T]) :- assert(P), rollback(T).
```

LISTING 6.3: Code for managing the initial state, verification, and rollback. The predicate `init(List)` is used to assert the facts contained in the initial state. The predicate `verify(List)` is used to infer the facts contained in the working memory. The predicate `rollback(List)` is used to perform a rollback of the facts asserted or withdrawn during planning, in case of failure of a recursion in the planner. These rules use the head/tail `[H|T]` notation for lists: the head of the list (its first element) is separated from the tail of the list (the rest of the elements); this notation is used to perform a particular operation sequentially on each element of the list.

```
bastian_plan(State, Goal, _, _, _) :- equal_set(State, Goal).
bastian_plan(State, Goal, Been_list, P, PNext) :-
    move(Name, Preconditions, Actions),
    verify(Preconditions),
    change_state(State, Actions, Child_state),
    (not(member_state(Child_state, Been_list)) ->
        (stack(Child_state, Been_list, New_been_list),
        add_to_queue(Name, P, PNew),
        bastian_plan(Child_state, Goal, New_been_list, PNew,
            PNext)
        -> (!); (rollback(Actions)))
    ; (rollback(Actions))), !.
```

LISTING 6.4: Prolog implementation of the `Bastian_Plan` predicate, which recursively searches for a sequence of actions to reach the goal state while avoiding cycles.

```
go(S,G,P) :- init(S), bastian_plan(S,G,[S],[],P).
```

LISTING 6.5: The first step is to initialise each fact of the initial state with `init(S)`, and then run the planner. Note the introduction of a variable `P` to ensure that the plan is returned in the form of a variable, for compatibility with the SWIProlog plugin in FANTASIA.

To exploit the potential of the inferential engine, a hierarchy of classes has been defined in Prolog, as shown in Listing 6.6.

```
object(X) :- tool(X).
tool(X) :- surface_tool(X).
tool(X) :- sharp_tool(X).
tool(X) :- mixing_tool(X).
object(X) :- food(X).
food(X) :- liquid_food(X).
object(X) :- container(X).
object(X) :- recipient(X).
recipient(X) :- cookware(X)
```

LISTING 6.6: Hierarchy of classes defined in Prolog. If `mixing_tool(X)` is present in the state, Prolog can infer that the variable X is also a tool, and therefore also a (generic) object.

This supports the integration of the Bastian Planner as follows: suppose the agent has a complete plan to prepare a sandwich and must decide how to instruct the interlocutor to complete the task with minimal instructions. The agent has a sequence of detailed steps, such as:

> *open the bread bag*, *take two slices of bread*, *take the peanut butter*, *open the jar of peanut butter*, *take a knife*, *spread the peanut butter on a slice of bread with the knife*, [...]

Instead of providing all this information, the agent optimises the instructions by combining several specific steps into one, assuming that the interlocutor knows the general process of sandwich preparation based on Background and Presupposed Knowledge(Chapter 4), reducing redundancy in the instructions. Therefore, instead of explicating *open the bread bag*, the system will simply mention *take two slices of bread* as the core part of the sandwich preparation. To replicate this process, tools integrated within FANTASIA have been leveraged.

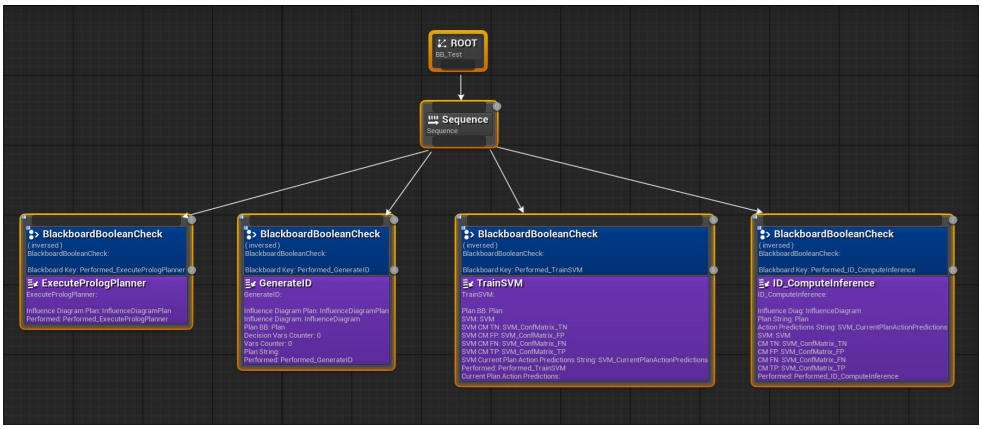

FIGURE 6.9: The Behaviour Tree was created to simulate the application. The tasks are executed in the specified sequence, from left to right. The `BlackboardBooleanCheck` is a simple mechanism used to prevent the sequence from being executed in a loop during the conducted tests.

Specifically it has been employed:

- Neo4j as a graph database;
- SWIProlog as an inferential engine;
- Influence Diagrams as a probabilistic graphical model;
- LibSVM as a module for realizing and training support vector machines for classification.

Since Blueprint is a visual scripting system, it can not directly display the tasks produced in this work. Therefore, it is represented only the Behavior Tree set-up for the simulation and the post-execution results in Figure 6.9 and 6.10 to demonstrate the feasibility of this approach, described as follows:

1. **The ExecutePrologPlanner module** executes the planning process in Prolog for the chosen test case and returns a string that outlines the instructions for assembling the Influence Diagram;
2. **GenerateID** analyzes the sequence generated by Prolog and assembles the Influence Diagram. In this phase, the structure of the decision-making model is created. Additionally, the action plan is isolated in Prolog and loaded into the graph database as the CURRENTPLAN test plan, with each action assigned a class of -1;

FIGURE 6.10: Simulation of the conversational agent Bastian. At the end of the reasoning phase illustrated here, the responses to the user were generated in natural language using ChatGPT. Bastian's response, displayed in green, suggests that to boil carrots, one should cut them, place them in a pan, pour in a small amount of water, and stir. He also noted that certain actions were omitted as they were considered common sense.

3. **TrainSVM** trains the SVM, calculating the node embedding in the graph database and returning a complete dataset. Each ACTION node is labelled 0, and each INSTRUCTION node is labelled 1. The CURRENTPLAN nodes with class -1 are separated from the rest, then the dataset is shuffled and balanced, dividing the data to obtain 3,015 ACTION nodes and 3,015 INSTRUCTION nodes. A 90%-10% ratio is used to obtain the training and test sets, which are then used to train the SVM. Finally, the nodes of the plan in question are predicted, and this information is suitably stored for management in the next task;

4. **ID_ComputeInference** fills the Conditional Probability Tables of the model's chance nodes, assigns the utilities, and conducts the inference.

The integral outputs printed during the debugging phase of the ID_ComputeInference task are reported in Listing 6.7. In Figure 6.9 is shown the Behaviour Tree and the results in Figure 6.10.

```
2_sayTAKE_0?: NO with p(ACTION) = 0.61
6_sayCUT_1?: YES with p(INSTRUCTION) = 1.00
10_sayWASH_2?: NO with p(ACTION) = 0.54
14_sayTAKE_3?: NO with p(ACTION) = 0.92
18_sayPUT-DOWN_4?: NO with p(ACTION) = 0.88
22_sayPOUR_1_5?: YES with p(INSTRUCTION) = 0.82
26_sayPOUR_2_6?: YES with p(INSTRUCTION) = 0.81
30_sayMIX_7?: YES with p(INSTRUCTION) = 1.00
34_sayWASH_8?: NO with p(ACTION) = 0.72
```

LISTING 6.7: List of moves with ACTION and INSTRUCTION labels. Each string shows the result of an analysis that aims to determine whether a given verb represents an action or an instruction, along with the probability that the classification is correct.

6.2.2 Results

To verify the applicability of the proposed planning, the system was tested on 10 different plans generated by Bastian's pre-inference process, which were then tested on human participants. Since the actions were shown in an unreadable format for the human users, ChatGPT was employed to transform the textual actions into natural language. For instance, an action such as *pour* was transformed as *Pour the carrot into the pan*. Figure 6.11 shows a test survey for one of the generated plans, where participants had to select at least one instruction from a list, each of which detailed the steps required to complete a recipe.The study involved 50 participants, who were asked to select the minimum set of instructions necessary for a user to follow the procedure for preparing a dish as planned, excluding any redundant actions (i.e. Commonsense). For each planning, the Dynamic Time Warping (DTW) metric [267] was employed to estimate the results. Specifically, it was estimated:

1. The agreement among participants;
2. The average DTW distance between participant pairs;
3. The sequence selected by the majority of participants;
4. The DTW distance between Bastian's estimated sequence and the sequence selected by the majority of participants.

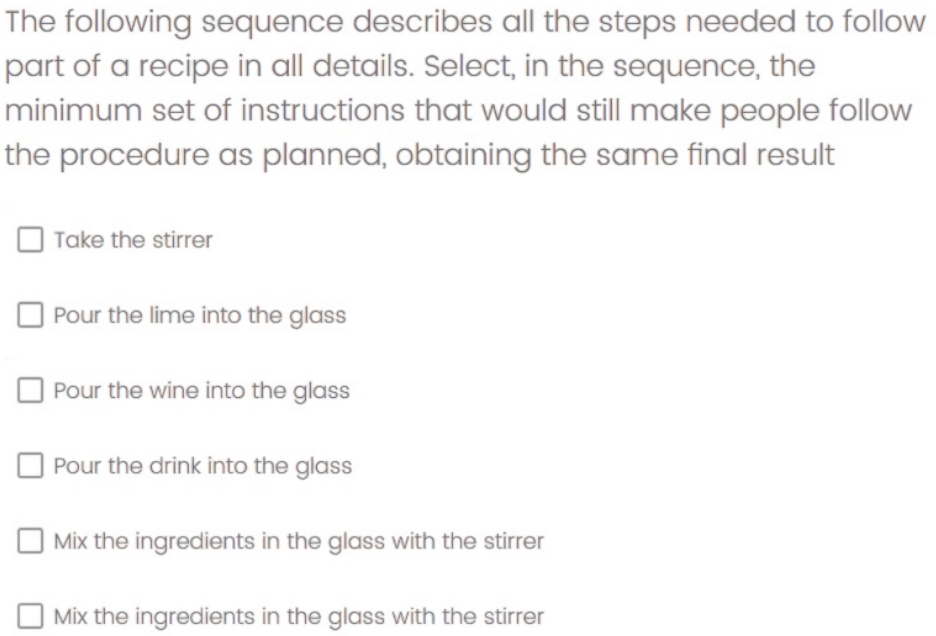

FIGURE 6.11: The user interface for the experiment required participants to select at least one option to move on to the next question, with a total of ten example plans.

Results are reported in Table 6.1, where each row represents one of the tested plans, and participants are denoted as P and the agent as B. For each plan, the *Agreement* parameter indicates the percentage of total agreement among participants on the sequences they generated; the *Sequence* parameter refers to the ideal sequence among all those generated by participants, with each action chosen according to a majority criterion. The *Sequence* parameter corresponds to the sequence chosen by the agent Bastian. The *DTW* parameter indicates the DTW distance between the sequence chosen by participants and the sequence chosen by Bastian. The *Avg. DTW* parameter represents the average DTW distance over the sequences between all pairs of participants. Lastly, the *Diff.* parameter refers to the difference between DTW and Avg.DTW. The DTW is a measure estimating the degree of closeness between the sequences. In this respect, the negative *Diff* suggests that Bastian's choices align more closely with what is generally perceived as true, compared to the average person.

Agreement	Sequence (P)	Sequence (B)	DTW(B,P)	Avg. DTW (P)	Diff.
96%	2,6,7,8	2,6,7,8	0.0	2.65	-2.65
96%	2,6,7,8,9	2,6,7,8,9,10	1.0	2.67	-1.67
97%	2,3,5	2,3,4,5,6	1.41	1.96	-0.54
89%	2,6,10,14,15	3,4,7,8,10,11,14,15	3.32	6.69	-3.37
92%	2,3,4,7,8,11,12,13,14,15	2,3,4,7,8,11,12,13,14,15	0.0	5.16	-5.16
92%	4,7,10,14,15	1,2,3,4,5,7,10,14,16	4.0	5.03	-1.03
92%	4,7,10,11	4,5,7,8,12,13	3.16	4.69	-1.53
86%	4,10,13,18,19	4,7,8,9,10,11,12,15,16,18,19,20	5.0	8.64	-3.64
83%	4,7,10,13,16,19,22,23	8,9,10,11,13,14,15,16,17,20,21,22,23,24	5.0	10.63	-5.63
95%	4,8,9,10	1,2,4,8,9,10	3.61	3.37	0.24

TABLE 6.1: Comparison between sequences selected by participants and sequences selected by Bastian, using Dynamic Time Warping.

Results show that participants exhibit a fairly broad consensus on what is deemed useful and what is not. In cases with a high degree of agreement (96%, 97%), the average DTW distance between the participants' sequences (Avg. DTW (P)) is generally low (e.g. 2.65 and 1.96). This indicates that the participants tend to converge on a common sequence, showing a shared perception of the best solution. When the agreement decreases (e.g. 83% or 86%), the average distance between participants increases significantly (10.63 and 8.64), indicating a greater variability in the decisions. This suggests that participants, in some cases, do not have a clear consensus on what the ideal sequence could be. When the Agreement is high (e.g. 96% or 97%), the difference (Diff) between DTW(B,P) and Avg. DTW(P) tends to be negative. This means that Bastian's choices are very close to those of the participants, suggesting that the model is well aligned with the human perception of the ideal sequence. For a lower level of agreement (e.g. 83%), there is a bigger difference between DTW(B,P) and Avg. DTW(P). Specifically, in the case with 83% agreement, there is a difference of -5.63, indicating that Bastian's choices are less aligned with those of the participants, probably due to the greater variability between human sequences. However, for the case with 95% agreement (last row), the difference is positive (0.24), suggesting that Bastian's choices in that specific case might be slightly less consistent with human choices than the average of the participants' sequences.

To summarize, results demonstrate that (i) the participants generally agree on what is useful and what is not, despite the presence of more intricate scenarios, such as the eighth and ninth action sequences. (ii) The deviations from the majority consensus are minimal, suggesting that Bastian's decisions, while not flawless, are also not incoherent. In this respect, (iii) Bastian appears to be more closely aligned with the overall sequence than the participants themselves.

6.3 Discussion

In this chapter we introduced FANTASIA, a plugin for the Unreal Engine designed to support the development of Embodied Conversational Agents. We have also introduced its interaction model, managing the dialogue as a physical system in which the participants exert forces on a network structure, pushing it through information exchanges towards an equilibrium state that exhibits the desired goal pattern. Concerning Undesirability, we saw that the Behaviour Tree for resolving undesirability has been implemented with a set of tasks that update the decision-making strategy. The inference engine can be used at runtime to plan a strategy and assemble probabilistic graphical models, which can then address the practical resolution of unwanted aspects associated with the conversation. As an example application of managing kitchen work collaboratively, it is here taken into account Bastian, an embodied conversational agent built in FANTASIA, who can plan a sequence of operations to solve a task, but does not have to carry it out alone and instead must instruct someone else. Assuming the complete plan is ready in the agent's control centre, it is necessary to decide which instructions to provide the interlocutor to complete the task with the fewest instructions. Within a broader theoretical framework provided in this work, this connects the capability to generate solutions to a problem, such as identifying all possible methods for accomplishing a task, with the process of selecting the optimal solution. The agent must decide whether to verbalize each step of the planning, balancing the need for efficiency with the necessity for accuracy. This involves evaluating the likelihood that an individual will perform a step in the process even if it is omitted, as well as assessing the probable consequences of choosing to skip a particular action. The methodological framework and the tools provided by FANTASIA make it possible to explore the relationship between linguistic models, numerical measures computed through graph data science, and the graphical structure of BNs. Indeed, in the FANTASIA architecture both modules are available and can be connected to implement such a mechanism. Moreover, FANTASIA supports the Neo4j database, so that information provided in this work can be employed for this purpose. Specifically, while logical engines do not consider the probability of a variable assuming a certain value, in most cases, probabilistic decision models do. On the other hand, decision models need prior probabilities to be estimated from

previous experience and compared with the current stimuli while also taking into account causal relationships. By considering the transition probability between steps, a decision model can estimate how probable it is for a person to perform the step in any case by recalling, through the database, how frequent it is for the two steps to come one after the other. In this respect, the Bastian Plan was developed. To verify its applicability, the system was tested on 10 different plans generated by Bastian's pre-inference process, which were then tested on human participants to deepen the investigation. Results demonstrated that the model seems to perform well in situations where there is a high level of agreement among participants, being aligned or even more consistent than the average of human choices. Moreover, in cases where there is less agreement among participants, his choices also tend to diverge more. Given a utility function rewarding the model for omitting the verbalisation of a step and penalising it for having the interlocutor miss a step, a risk of omitting each step can be computed. This leads to a sequence of decisions to omit or not every step, trying to find the best balance between the two. Also, by observing what happens during the interaction and checking if a step is performed or not when it is omitted from the instructions, it is possible to take more informed decisions as the interaction continues, thus avoiding the problem of deciding everything beforehand and taking into account all possible outcomes in an up-to-date way.

CHAPTER

SEVEN

CONCLUSION AND FUTURE WORK

A key aspiration of AI is to integrate conceptual and behavioral knowledge in machines, bridging the gap in the human-machine interaction. In this regard, CSK is a topic of utmost importance, recognised as a central challenge in the AI and NLP fields, and continues to puzzle researchers. In this respect, most of the work conducted by Gary Marcus and Ernest Davis has been focused on CS since 2015 [68], highlighting the need for machines to possess CS to reach human-level intelligence. As highlighted in a recent post on Substack published by the authors [1], AI systems continue to grapple with difficulties in real-world scenarios as they can only reason and plan when they have access to complete and detailed maps of the physical world, which can be hard to obtain, as already discussed in Davis et al., [67]. Although significant progress has been made in recent years, particularly with the rise of LLMs, many issues remain unresolved. The authors emphasize that AI is still far from reaching a stage where it can be considered truly developed, as highlighted in the following passage:

"[...] Our view is that they [AI systems] will not work well until major progress is made in common sense, which itself may require wholesale revisions in how AI is approached."

As it has been pointed out, the inherently subjective and context-dependent nature of CS has long made it difficult to study this phenomenon in the AI field, partly because

[1]Gary Marcus and Ernest Davis. January 5th, 2025: AI still lacks 'common' sense, 70 years later - What's obvious to people still isn't always obvious to machines: https://garymarcus.substack. com/p/ai-still-lacks-common-sense-70-years [last visited on: 10/01/2025]

it is challenging to apply standard experimental techniques to investigate it. To address this problem, my research initially focused on finding an adequate definition to frame the concept of CS - and, specifically, of CSK -, exploring potential connections with the field of cognitive linguistics (Chapter 2 and 3). This investigation appears to have been "foresightful", as a recent survey by Do et al. [76] reveals no clear, well-established definition of CSK from a linguistic perspective. Building on linguistic theories, I aimed to explore the distinctions between knowledge consolidated in linguistic theory (i.e. Common Ground and Encyclopedic Knowledge) and that of CSK, providing a foundation for further theoretical exploration of these aspects. In this respect, I proposed to conceptualize this knowledge as a dynamic process rather than a static representation of facts (Chapter 4). This approach builds upon the Saba's idea [264], which states that "ordinary language is the best-known theory we have of everyday cognition". Therefore, developing a model that extracts underlying processes directly from linguistic data seemed more human-like. To conduct the analysis, the cooking domain was taken into account for (i) its ubiquitous familiarity and (ii) the presence of implicit systematic chains of actions in the cooking process. Drawing from cognitive linguistics (Chapter 2), information was divided into three macro categories (Foreground Knowledge, Background Knowledge and Presupposed Knowledge) and a three-level analysis was proposed to facilitate clearer understanding and more effective linguistic data analysis (Chapter 5).

Therefore, this work presented two key contributions, discussed as follows:

- **C1**: *A linguistic-methodological framework for investigating the semantic dimensions of implicit and explicit information and distribution along the dialogue flow.*

  In Section 5.1, the main aim was to recover semantic information from a surface layer (i.e. foreground knowledge), focusing on *Semantic Knowledge*. Leveraging the FrameNet lexical base, semantic domain information has been identified to extract the positions of explicit information from CookDial corpus within the dialogue flow. Results suggest that the frame distributions reflect the natural flow of a culinary task, where initial steps involve preparing ingredients and manipulating temperature, while later stages focus on cooking food, monitoring progress, modelling the shape and finalising the process, offering an overview of actions performed before others. In Section 5.3, the main goal

was to determine whether there were any linguistic differences between explicit and implicit information. Therefore, to detect linguistic aspects that distinguish verbalised actions from those left implicit, a descriptive statistics analysis of the Epic Kitchens data has been conducted, with a particular focus on verb classes 0 (take) and 1 (put). Analysing the syntactic structures of the sentences along with their semantic roles made it evident that certain structures must inevitably be made explicit to understand the task at hand, while others can be omitted. Thanks to this linguistic analysis, it was possible to describe the semantic particularities that highlight the differences between the two types of information and their distribution along the dialogue flow.

- **C2**: *Employing graph databases to enable cross-domain analysis and probabilistic estimation, allowing for the identification of co-occurring actions associated with specific items.*

  In Section 5.2, it has been build the cooking domain knowledge base containing all the interconnected information, establishing what has been defined as *Background Knowledge*. Therefore, the procedure for integrating the Recipe1M+, FlavorDB and Epic Kitchens data into Neo4J to build the knowledge base was explained. This made it possible to apply a cross-domain analysis over the frequency of actions performed on ingredients belonging to specific categories within FlavorDB, providing information on the categorical distribution of ingredients on which certain actions are performed. Results demonstrate a strong relationship between the object and the types of manipulation applied to them. As that information is well established in the speaker's mind by experience, it leads us to state that those can be omitted from communicative exchanges, as they belong to the general understanding of the world. In this regard, a co-occurrence matrix of verbs and nouns was extracted from Epic-Kitchens and integrated into the database employing markov chains. Results shown that the recovery of implicit information is possible by retrieving it through a probability estimation within the graph.

In other words, knowing if an object can be cut (*Background Knowledge*, Section 5.2) as well as the knowledge that an object cannot be cut without a sharp tool capable of cutting it, is already shared in the speaker's mind. Therefore, following a logic flow,

such as (i) taking the tool (non-core) and (ii) cutting the object (core), the speaker would only make explicit the core information (*Semantic Knowledge*, Section 5.1). The listener, who shares the same knowledge of the world, can retrieve the non-core information (*Presupposed Knowledge*, Section 5.3), through a probabilistic estimate.

As mentioned at the beginning of this work, the aim pursued here was to deepen the CSK structure for the implementation of conversational agents, starting from a linguistic perspective. Dialogue systems are specifically designed to enable safe, trustworthy, and personalised interactions, addressing key aspects of human-like communication and cognitive engagement. In this respect, there is a need for both communicative competence inherent to reasoning and sufficient coverage of the breadth of CS concepts for the retrieval of language understanding, perception, similarity, and other cognitive functions. We already mentioned that there are two main types of dialogue systems, such as (i) Task-oriented Dialogue Systems (TOD), specifically designed to efficiently manage task-oriented conversations, helping users achieve specific objectives by detecting user intentions, tracking dialogue states, executing appropriate actions, and providing relevant responses, involving a sequence of actions or sub-steps that are necessary to accomplish the main objective. (ii) Open-domain Dialogue Systems (OOD) are intended for open-domain interactions, facilitating free-flowing conversations across a wide array of topics by directly mapping the dialogue context to the response, without a predefined task or goal. Between 2019 and 2022, the integration of some aspects of TOD and OOD, facilitated by advancements in deep learning and large-scale language datasets, has led to the emergence of Pre-trained Language Models (PTLMs), then transformed into Large Language Models (LLMs), transforming the landscape of dialogue systems. Unlike basic pre-programmed chatbots, LLMs operate as generative models possessing considerable inferential capacity and extensive reservoirs of knowledge, interacting with external contextual knowledge after deployment. While how "human-like" the state-of-the-art LLMs are (cognitive plausibility) has not comprehensively justified [318], Goldstein et al. [119] provide empirical evidence that the human brain and GPT-2 share fundamental computational principles in processing natural language. Both are engaged in continuous next-word prediction, and they represent words as a function of the previous context. Against this background, the study conducted by Cong [62] investigates the cognitive plausibility of language models by examining their performance in understanding pragmatically enriched meanings, which are implied or presupposed

among most people to convey their intentions. The work analyzed neural language models' understanding of commonsense pragmatics through human behavioural and neurophysiological data to draw conclusions based on predictive responses in context, making them well-suited to test word-prediction models such as BERT in natural settings. The findings suggest that GPT-3's performance [44] was mostly at chance in the psycholinguistic tasks and DistillBERT [270] had some understanding of the intent that is shared among most people, as reflected in the usage of conversational implicatures and presuppositions. Whether fine-tuning improves its performance to the human level depends on the type of CS reasoning involved. In future work, a more in-depth study of all cognitive and linguistic aspects should be carried out to obtain a more comprehensive theoretical framework that will enable further experimentation. I would argue that this idea is reinforced by the statement expressed in Marcus's 2019 book [196], which states that:

"[...] The only way out of this mess is to get cracking on building machines equipped with common sense, cognitive models, and powerful tools for reasoning. Together, these can lead to deep understanding, itself a prerequisite for building machines that can reliably anticipate and evaluate the consequences of their own actions."[p.199]

This is in line with the FANTASIA dialogue architecture (Chapter 6), which links graph-based resource analysis procedures grounded in linguistic theory with the technological tools needed to implement conversational agents using the Unreal Engine. As we already saw, FANTASIA keeps graph-based knowledge, decision models, and logical capabilities separate from the domain of LLMs, which are instead used as natural language generators that act on directives generated by other AI modules. In the specific case of identifying CS instructions that can be omitted to follow a principle of maximum utility, the role of logical engines within the proposed architecture is to dynamically generate decision models that facilitate reasoning under uncertainty. In managing kitchen work collaboratively, it has been taken into account the case of Bastian, an embodied conversational agent built in FANTASIA, who can plan a sequence of operations to solve a task collaboratively. Instead of providing a sequence of detailed steps to complete a task (e.g. preparing a sandwich), the agent would automatically optimize the instructions by combining several specific steps into one, assuming that the interlocutor knows the general process of sandwich preparation

based on his experience of the world. This approach reduces redundancy in the instructions, improving the fluency of the interaction. To verify its applicability, the system was tested on 10 different plans generated by Bastian's pre-inference process, which were then tested on human participants to deepen the investigation. Results demonstrated that the model seems to perform well in situations where there is a high level of agreement among participants, being aligned or even more consistent than the average of human choices. Moreover, in cases where there is less agreement among participants, his choices also tend to diverge more. This lack of agreement among participants closely reflects what was already observed in the experiment conducted by Whiting and Watt [324]. Although the work conducted here focused on the culinary domain, which typically exhibits greater agreement also according to their case study, differences can still arise, especially when dealing with a long sequence of instructions. As evidenced by the results, participants tend to disagree when navigating lengthy instruction sets, a pattern mirrored in the choices made by the system. In conclusion, the observed non-agreement is consistent with the system's behaviour, validating the implemented plan. However, further verifications will be necessary, particularly regarding these extensive action lists, to identify the most efficient strategies for completing the task. Moreover, given the normative nature of texts such as recipes, future work will involve testing different text typologies and, if necessary, refining and expanding the proposed three-analysis scheme. Another aspect for future exploration involves conducting comparative evaluations with traditional static CSK models, which would further highlight the strengths of the dynamic process-based framework.

In conclusion, within a broader theoretical framework provided in this work, this connects the capability to generate solutions to a problem, such as identifying all possible methods for accomplishing a task, with the process of selecting the optimal solution. This way, the human-machine interaction will be facilitated, avoiding overwhelming the user with unnecessary information and reaching the communicative goal. In this respect, Common Sense emphasises - to some extent - that a minimal and concise approach can be more effective, in contrast to other approaches that employ millions of data trying to make machines increasingly more "efficient", at the expense of their quality. In other words:

less is **more**.

BIBLIOGRAPHY

[1] Ralph Adolphs. “Cognitive Neuroscience of Human Social Behaviour”. In: *Nature Reviews Neuroscience*. Vol. 4. 3. London, UK: Nature Publishing Group, 2003, pp. 165–178. DOI: https://doi.org/10.1038/nrn1056.

[2] Colin Aitken and Dimitris Mavridis. “Reasoning under uncertainty”. In: *BMJ Ment Health* 22.1 (2019), pp. 44–48.

[3] Jens Allwood. “Dimensions of embodied communication—towards a typology of embodied communication”. In: *Embodied Communication in Humans and Machines*. Ed. by Manuela Lenzen Ipke Wachsmuth and Günther Knoblich. Oxford: Oxford University Press, Sept. 2008. Chap. 12.

[4] Jens Allwood. “A Framework for Studying Human Multimodal Communication”. In: *Coverbal Synchrony in Human-Machine Interaction*. Ed. by Nick Campbell Matej Rojc. 1st ed. Vol. 17. Boca Raton, FL: CRC Press, Taylor & Francis Group, 2013. Chap. 2.

[5] Patrícia Amaral, Chris Cummins, and Napoleon Katsos. “Experimental Evidence on the Distinction Between Foregrounded and Backgrounded Meaning”. In: *Proceedings of the 2011 ESSLLI Workshop on Projective Content*. Columbus, OH: Department of Linguistics, OSU, 2011, pp. 1–7.

[6] Junia Anacleto et al. “Can Common Sense uncover cultural differences in computer applications?” In: *Artificial Intelligence in Theory and Practice*. Ed. by Max Bramer. Boston, MA: Springer US, 2006, pp. 1–10.

[7] József Andor. “Discussing Frame Semantics: The State of the Art—An Interview with Charles J. Fillmore”. In: *Review of Cognitive Linguistics*. Vol. 8. 1. John Benjamins Publishing, 2010, pp. 157–176. DOI: 10.1075/rcl.8.1.06and.

[8] Renzo Angles et al. “Foundations of Modern Query Languages for Graph Databases”. In: *ACM Comput. Surv.* 50.5 (Sept. 2017). DOI: 10.1145/3104031.

[9] Fabian Anicker and Florian Golo Flaßhoff. “Common-sense attributions of AI agency: Evidence from an experiment with ChatGPT”. In: *AI and Common Sense*. Ed. by Bernard Schiele Martin W. Bauer. 1st ed. London, UK: Routledge, 2024, pp. 179–194.

[10] Ian Apperly. *Mindreaders: the cognitive basis of" theory of mind"*. Psychology Press, 2010.

[11] Jay David Atlas. “On the Semantics of Presupposition and Negation: An Essay in Philosophical Logic and the Foundations of Linguistics”. PhD thesis. Princeton University, 1976.

[12] Sören Auer et al. “Dbpedia: A nucleus for a web of open data”. In: *The Semantic Web. ISWC ASWC 2007*. Ed. by K. Aberer et al. Vol. 4825. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 722–735.

[13] John L. Austin. *How to Do Things with Words*. Ed. by Marina Sbisá and J. O. Urmson. Oxford: Clarendon Press, 1962.

[14] Franz Baader and Werner Nutt. “Basic description logics”. In: *The description logic handbook: theory, implementation, and applications*. 2003, pp. 43–95.

[15] Chris L. Baker, Rebecca Saxe, and Joshua B. Tenenbaum. “Action understanding as inverse planning”. In: *Cognition* 113.3 (2009). Reinforcement learning and higher cognition, pp. 329–349. DOI: 10.1016/j.cognition.2009.07.005.

[16] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. “The Berkeley FrameNet Project”. In: *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*. Vol. 1. Montreal, Quebec, Canada: Association for Computational Linguistics, 1998, pp. 86–90.

[17] Vevake Balaraman, Seyedmostafa Sheikhalishahi, and Bernardo Magnini. “Recent neural methods on dialogue state tracking for task-oriented dialogue systems: A survey”. In: *Proceedings of the 22nd Annual Meeting of the*

Special Interest Group on Discourse and Dialogue. Singapore and Online: Association for Computational Linguistics, 2021, pp. 239–251. DOI: 10.18653/v1/2021.sigdial-1.25.

[18] Yejin Bang et al. "A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity". In: *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*. Vol. Volume 1: Long Papers. Nusa Dua, Bali: Association for Computational Linguistics, 2023. DOI: 10.18653/v1/2023.ijcnlp-main.45.

[19] Chitta Baral. *Knowledge representation, reasoning and declarative problem solving*. Cambridge university press, 2003.

[20] Antonio Barcelona and Javier Valenzuela. "An overview of cognitive linguistics". In: *Cognitive Linguistics: Convergence and Expansion*. Ed. by Mario Brdar, Stefan Th. Gries, and Milena Žic Fuchs. John Benjamins Publishing Company, 2011, pp. 17–44.

[21] Jonathan Barnes. *Complete Works of Aristotle, Volume 1: The Revised Oxford Translation*. Princeton: Princeton University Press, 1984.

[22] Lawrence W. Barsalou. "Perceptual symbol systems". In: *Behavioral and Brain Sciences*. Vol. 22. 4. Cambridge: Cambridge University Press, 1999, pp. 577–609. DOI: 10.1017/s0140525x99002149.

[23] Frederic C. Bartlett and Walter Kintsch. *Remembering: A study in experimental and social psychology*. 2nd ed. Cambridge: Cambridge University Press, 1995.

[24] Laura Bartlett. "Common sense, artificial intelligence and psychology". In: *AI and Common Sense*. Ed. by Bernard Schiele Martin W. Bauer. London,UK: Routledge, 2024. Chap. 6, pp. 82–96.

[25] Martin W Bauer. "AI with common sense: What concept of common sense?" In: *AI and Common Sense*. Ed. by Bernard Schiele Martin W. Bauer. London, UK: Routledge, 2024, pp. 13–29.

[26] David I. Beaver, Bart Geurts, and Kristie Denlinger. “Presupposition”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta and Uri Nodelman. Fall 2024. Metaphysics Research Lab, Stanford University, 2024.

[27] David Ian Beaver. “Presupposition”. In: *Handbook of Logic and Language*. Ed. by Johan van Benthem and Alice ter Meulen. Amsterdam: North-Holland, 1997. Chap. 17, pp. 939–1008.

[28] Michael Beetz et al. “AI reasoning methods for robotics”. In: *Springer Handbook of Robotics* (2016), pp. 329–356.

[29] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. “A Neural Probabilistic Language Model”. In: *Journal of Machine Learning Research*. Vol. 3. MIT Press, Jan. 2000, pp. 932–938.

[30] Luisa Bentivogli et al. “The Fifth PASCAL Recognizing Textual Entailment Challenge.” In: *Proceedings of the Second Text Analysis Conference (TAC 2009)*. Gaithersburg, Maryland, USA: NIST, 2009.

[31] Michael K Bergman. “A common sense view of knowledge graphs”. In: *Adaptive Information, Adaptive Innovation, Adaptive Infrastructure Blog* (2019). URL: http://www.mkbergman.com/2244/a-common-sense-view-of-knowledge-graphs.

[32] Brent Berlin and Paul Kay. *Basic Color Terms: Their Universality and Evolution*. Berkeley & Los Angeles: University of California Press, 1969.

[33] Tim Berners-Lee, James Hendler, and Ora Lassila. “Web Semantic”. In: *Scientific American* 284.5 (2001), pp. 34–43.

[34] Patrick Blackburn, Johannes Bos, Kristina Striegnitz, et al. *Learn prolog now!* Vol. 7. 7. College Publications London, 2006.

[35] Andreas Blumauer. “From taxonomies over ontologies to knowledge graphs”. In: *Semantic Web Company* (2014). URL: https://semantic-web.com/from-taxonomies-over-ontologies-to-knowledge-graphs/.

[36] Daniel G. Bobrow et al. “GUS, a frame-driven dialog system”. In: *Artificial Intelligence* 8.2 (1977), pp. 155–173. DOI: https://doi.org/10.1016/0004-3702(77)90018-2.

[37] Piero Andrea Bonatti et al. "Knowledge graphs: New directions for knowledge representation on the semantic web (Dagstuhl Seminar 18371)". In: *Dagstuhl Reports*. Vol. 8. 9. Saarbrücken/Wadern: Schloss Dagstuhl-Leibniz-Zentrum für Informatik, Dagstuhl Publishing, 2019.

[38] Antoine Bordes et al. "Learning structured embeddings of knowledge bases". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 25. 1. 2011, pp. 301–306.

[39] Ali Borji. *A Categorical Archive of ChatGPT Failures*. 2023. arXiv: 2302.03494. URL: https://arxiv.org/abs/2302.03494.

[40] Ronald Brachman and Hector Levesque. *Knowledge representation and reasoning*. San Francisco, CA: Elsevier, 2004. DOI: 10.1016/B978-1-55860-932-7.X5083-3.

[41] Martina Di Bratto. "Journey through Argumentation-based Dialogue: from theoretical to computational models for the implementation of Conversational Recommender Systems". PhD thesis. Università degli Studi di Napoli Federico II, 2024.

[42] Jacob Bronowski. *The Common Sense of Science*. Harvard University Press, 1953 [1978].

[43] Rodney A Brooks. "Intelligence without representation". In: vol. 47. 1-3. Elsevier, 1991, pp. 139–159.

[44] Tom Brown et al. "Language models are few-shot learners". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20)*. Vol. 33. Vancouver, BC, Canada: Curran Associates Inc., 2020, pp. 1877–1901.

[45] Jos de Bruijn et al. "D4. 2.1 state-of-the-art-survey on ontology merging and aligning v1". In: *SEKT Project deliverable D* 4 (2004), pp. 3–94.

[46] Joan L. Bybee. "From usage to grammar: The mind's response to repetition". In: *Language:Journal of the Linguistic Society of America* 82.4 (2006), pp. 711–733.

[47] Ruth MJ Byrne. "Suppressing valid inferences with conditionals". In: *Cognition*. Vol. 31. 1. Elsevier, 1989, pp. 61–83.

[48] Erik Cambria et al. “Common sense computing: From the society of mind to digital intuition and beyond”. In: *Biometric ID Management and Multimodal Communication*. Ed. by Julian Fierrez et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 252–259. ISBN: 978-3-642-04391-8.

[49] Erik Cambria et al. “Isanette: A common and common sense knowledge base for opinion mining”. In: *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE. 2011, pp. 315–322. DOI: 10.1109/ICDMW.2011.106.

[50] Erik Cambria et al. “SenticNet”. In: *Sentic computing: a common-sense-based framework for concept-level sentiment analysis* (2015), pp. 23–71. DOI: 10.1007/978-3-319-23654-4_2.

[51] Hakki C Cankaya and Dan Moldovan. “Method for extracting commonsense knowledge”. In: *Proceedings of the Fifth International Conference on Knowledge Capture*. New York, NY, USA: Association for Computing Machinery, 2009, pp. 57–64. DOI: 10.1145/1597735.1597746.

[52] Michael J Carter. “The hermeneutics of frames and framing: An examination of the media’s construction of reality”. In: vol. 3. 2. SAGE Publications Sage CA: Los Angeles, CA, 2013. DOI: 10.1177/2158244013487915.

[53] Yupeng Chang et al. “A Survey on Evaluation of Large Language Models”. In: *ACM Transactions on Intelligent Systems and Technology* 15.3 (2024), pp. 1–45. DOI: 10.1145/3641289.

[54] Eugene Charniak. *Statistical language learning*. Cambridge, Mass: MIT press, 1996.

[55] Agnese Chiatti, Enrico Motta, and Enrico Daga. “Towards a Framework for Visual Intelligence in Service Robotics: Epistemic Requirements and Gap Analysis”. In: *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning*. Sept. 2020, pp. 905–916. DOI: 10.24963/kr.2020/93.

[56] Noam Chomsky. *Lectures on Government and Binding*. Mouton: The Gruyter, 1981.

[57] Noam Chomsky. *Knowledge of language: Its nature, origin, and use*. New York: Praeger, 1986.

[58] Philipp Cimiano and Heiko Paulheim. "Knowledge graph refinement: A survey of approaches and evaluation methods". In: *Semantic Web* 8.3 (Jan. 2017), 489–508. DOI: 10.3233/SW-160218.

[59] Herbert H Clark. *Using language*. Cambridge University Press, 1996.

[60] Herbert H. Clark and Susan E. Brennan. "Grounding in Communication". In: *Perspectives on Socially Shared Cognition*. Ed. by Lauren Resnick et al. American Psychological Association, 1991, pp. 13–1991.

[61] Herbert H. Clark and Edward F. Schaefer. "Contributing to discourse". In: *Cognitive Science* 13.2 (1989), pp. 259–294. DOI: 10.1016/0364-0213(89)90008-6.

[62] Yan Cong. "Psycholinguistic diagnosis of language models' commonsense reasoning". In: *Proceedings of the First Workshop on Commonsense Representation and Reasoning (CSRR 2022)*. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 17–22. DOI: 10.18653/v1/2022.csrr-1.3.

[63] Eugenio Coseriu. "Linguistic competence: What is it really?" In: *The Modern Language Review* 80.4 (1985), pp. 25–35. DOI: 10.2307/3729050.

[64] William Croft. "Linguistic evidence and mental representations". In: *Cognitive Linguistics* 9.2 (1998), pp. 151–174. DOI: doi:10.1515/cogl.1998.9.2.151.

[65] Ido Dagan, Oren Glickman, and Bernardo Magnini. "The PASCAL recognising textual entailment challenge". In: *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*. Berlin, Heidelberg: Springer-Verlag, 2005, 177–190. DOI: 10.1007/11736790_9.

[66] Dima Damen et al. "Scaling Egocentric Vision: The EPIC-KITCHENS Dataset". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 720–736. DOI: 10.1007/978-3-030-01225-0_44.

[67] Ernest Davis. "Logical formalizations of commonsense reasoning: a survey". In: *Journal of Artificial Intelligence Research* 59 (2017), pp. 651–723. DOI: 10.1613/jair.5339.

[68] Ernest Davis and Gary Marcus. “Commonsense reasoning and commonsense knowledge in artificial intelligence”. In: *Communications of the ACM* 58.9 (Aug. 2015), pp. 92–103. DOI: 10.1145/2701413.

[69] Ferdinand De Saussure. “Course in general linguistics”. In: *Literary theory: An anthology* 2 (2004). Ed. by Julie Rivkin and Michael Ryan, pp. 59–71.

[70] Robert M DeKeyser. “Cognitive-psychological processes in second language learning”. In: *The handbook of language teaching* (2009). Ed. by Michael H. Long and Catherine J. Doughty, pp. 119–138. DOI: 10.1002/9781444315783.ch8.

[71] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. 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)*. 2019, pp. 4171–4186.

[72] Martina Di Bratto et al. “Dialogue Analysis with Graph Databases: Characterising Domain Items Usage for Movie Recommendations.” In: *CLiC-it*. Torino: Accademia University Press, 2022. DOI: 10.4000/books.aaccademia.10417.

[73] Maria Di Maro. “" Shouldn’t I use a polar question?" Proper Question Forms Disentangling Inconsistencies in Dialogue Systems.” PhD thesis. 2021.

[74] Maria Di Maro, Antonio Origlia, and Francesco Cutugno. “Cutting melted butter? Common Ground inconsistencies management in dialogue systems using graph databases”. In: *IJCoL. Italian Journal of Computational Linguistics* 7.7-1, 2 (2021), pp. 157–190.

[75] Felix Dietze et al. “An open-source object-graph-mapping framework for Neo4j and Scala: Renesca”. In: *Availability, Reliability, and Security in Information Systems*. Springer International Publishing. Cham, 2016, pp. 204–218. DOI: 10.1007/978-3-319-45507-5_14.

[76] Quyet V Do et al. “What Really is Commonsense Knowledge?” In: *ArXiv Preprint* (2024). DOI: arXiv:2411.03964.

[77] Xin Dong et al. “Knowledge vault: A web-scale approach to probabilistic knowledge fusion”. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2014, pp. 601–610. DOI: 10.1145/2623330.2623623.

[78] Jon Doyle. “A truth maintenance system”. In: *Artificial Intelligence* 12.3 (1979), pp. 231–272. ISSN: 0004-3702. DOI: https://doi.org/10.1016/0004-3702(79)90008-0. URL: https://www.sciencedirect.com/science/article/pii/0004370279900080.

[79] Mauro Dragoni, Soujanya Poria, and Erik Cambria. “OntoSenticNet: A commonsense ontology for sentiment analysis”. In: *IEEE Intelligent Systems* 33.3 (2018), pp. 77–85. DOI: 10.1109/MIS.2018.033001419.

[80] Georgios Drakopoulos et al. “On converting community detection algorithms for fuzzy graphs in Neo4j”. In: *ArXiv* (2016). DOI: 10.48550/arXiv.1608.02235.

[81] Zhengxiao Du et al. “GLM: General Language Model Pretraining with Autoregressive Blank Infilling”. In: *ArXiv* (2021). DOI: 10.48550/arXiv.2103.10360.

[82] Gaspard Ducamp, Christophe Gonzales, and Pierre-Henri Wuillemin. “aGrUM/pyAgrum: a toolbox to build models and algorithms for Probabilistic Graphical Models in Python”. In: *Proceedings of the 10th International Conference on Probabilistic Graphical Models*. Ed. by Manfred Jaeger and Thomas Dyhre Nielsen. Vol. 138. Proceedings of Machine Learning Research. Skørping, Denmark, 2020, pp. 609–612. URL: https://proceedings.mlr.press/v138/ducamp20a.html.

[83] Myroslava O Dzikovska et al. “SemEval-2013 Task 7: The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge”. In: *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Ed. by Suresh Manandhar and Deniz Yuret. Atlanta, Georgia, USA: Association for Computational Linguistics, June 2013, pp. 263–274. URL: https://aclanthology.org/S13-2045/.

[84] Lisa Ehrlinger and Wolfram Wöß. "Towards a definition of knowledge graphs." In: *Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16) co-located with the 12th International Conference on Semantic Systems (SEMANTiCS 2016)*. Ed. by Michael Martin, Martí Cuquet, and Erwin Folmer. Vol. 1695. CEUR Workshop Proceedings. Leipzig, Germany: CEUR-WS.org, 2016. URL: https://ceur-ws.org/Vol-1695/paper4.pdf.

[85] Robert M Entman. "Framing: Toward clarification of a fractured paradigm". In: *Journal of Communication* 43.4 (Feb. 1993), pp. 51–58. DOI: 10.1111/j.1460-2466.1993.tb01304.x.

[86] Jérôme Euzenat. "Uncertainty in crowdsourcing ontology matching". In: *Proc. 8th ISWC workshop on ontology matching (OM)*. Sydney, Australia, Oct. 2013, pp. 221–222.

[87] Vyvyan Evans. "The Structure of Time: Language, Meaning and Temporal Cognition". In: *Human Cognitive Processing - Cognitive Foundations of Language Structure and Use*. Ed. by Klaus-Uwe Panther and Linda L. Thornburg. Amsterdam: John Benjamins Publishing Company, 2004, pp. x,286. DOI: 10.1075/hcp.12.

[88] Vyvyan Evans. "Lexical concepts, cognitive models and meaning-construction". In: *Cognitive Linguistics* 17.4 (2006), pp. 491–534. DOI: 10.1515/COG.2006.016.

[89] Vyvyan Evans. "Cognitive linguistics". In: *Wiley interdisciplinary reviews: cognitive science* 3.2 (2012), pp. 129–141. DOI: 10.1002/wcs.1163.

[90] Vyvyan Evans, Benjamin K Bergen, and Jörg Zinken. "The cognitive linguistics enterprise: An overview". In: *The cognitive linguistics reader*. Ed. by Vyvyan Evans, Benjamin K Bergen, and Jörg Zinken. Advances in cognitive linguistics. United Kingdom: Equinox Publishing Ltd, 2007, pp. 2–36. ISBN: 9781845531102.

[91] Vyvyan Evans and Melanie Green. *Cognitive linguistics: An introduction*. New York: Routledge, 2006. DOI: 10.4324/9781315864327.

[92] Michael Färber and Achim Rettinger. "A Statistical Comparison of Current Knowledge Bases." In: *Joint Posters and Demos Track of 11th International Conference on Semantic Systems, Posters and Demos@SEMANTiCS 2015 and 1st Workshop on Data Science: Methods, Technology and Applications, DSci 2015 - co-located with the 11th International Conference on Semantic Systems, SEMANTiCS 2015*. Ed. by A. Polleres. Vol. 1481. CEUR Workshop Proceedings. Vienna; Austria: RWTH Aachen, 2015, pp. 18–21. DOI: 10.5445/IR/1000092934.

[93] Michael Färber et al. "Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO". In: *Semantic Web* 9.1 (Jan. 2018), pp. 77–129. DOI: 10.3233/SW-170275.

[94] Gilles Fauconnier. "Mappings in thought and language". In: *Cambridge University Press* (1997).

[95] Christina Feilmayr and Wolfram Wöß. "An analysis of ontologies and their success factors for application to business". In: *Data & Knowledge Engineering* 101 (2016), pp. 1–23. DOI: 10.1016/j.datak.2015.11.003.

[96] Leonardo Fernandino et al. "Concept representation reflects multimodal abstraction: A framework for embodied semantics". In: *Cerebral cortex* 26.5 (Mar. 2016), pp. 2018–2034. DOI: 10.1093/cercor/bhv020.

[97] C. J. Fillmore. "Frame semantics". In: *Linguistics in the Morning Calm*. Ed. by The Linguistic Society of Korea. Seoul: Hanshin, 1982, pp. 111–137.

[98] Charles J. Fillmore. "The Case for Case, Dins". In: *Universals in Linguistic Theory*. Ed. by Emmon W. Bach and Robert Thomas Harms. New York, NY, USA: Holt, Rinehart, and Winston, 1968, pp. 11–88.

[99] Charles J Fillmore. "Syntactic intrusions and the notion of grammatical construction". In: *Proceedings of the Eleventh Annual Meeting of the Berkeley Linguistics Society*. 1985, pp. 73–86. DOI: 10.3765/bls.v11i0.1913.

[100] Charles J Fillmore and Collin Baker. "A Frames Approach to Semantic Analysis". In: *The Oxford Handbook of Linguistic Analysis*. Ed. by Bernd Heine and Heiko Narrog. Oxford University Press, Dec. 2009. DOI: 10.1093/oxfordhb/9780199544004.013.0013.

[101] Charles J Fillmore et al. “Frame semantics and the nature of language”. In: *Annals of the New York Academy of Sciences: Conference on the origin and development of language and speech*. Vol. 280. 1. New York. 1976, pp. 20–32. DOI: 10.1111/j.1749-6632.1976.tb25467.x.

[102] Sarah E Finch and Jinho D Choi. “ConvoSense: Overcoming Monotonous Commonsense Inferences for Conversational AI”. In: *Transactions of the Association for Computational Linguistics* 12 (May 2024), pp. 467–483. DOI: 10.1162/tacl_a_00659.

[103] Gonzalo Flórez-Puga et al. “Query-enabled behavior trees”. In: *IEEE Transactions on Computational Intelligence and AI in Games* 1.4 (2009), pp. 298–308. ISSN: 1943-068X. DOI: 10.1109/TCIAIG.2009.2036369.

[104] Valentina Focaroli and Jana M Iverson. “Children’s object manipulation: a tool for knowing the external world and for communicative Development”. In: *The Hand: Perception, Cognition, Action*. Ed. by Marta Bertolaso and Nicola Di Stefano. Cham: Springer International Publishing, 2017, pp. 19–27. DOI: 10.1007/978-3-319-66881-9_2.

[105] Nadime Francis et al. “Cypher: An evolving query language for property graphs”. In: *Proceedings of the 2018 international conference on management of data*. SIGMOD ’18. Houston, TX, USA: Association for Computing Machinery, 2018, pp. 1433–1445. DOI: 10.1145/3183713.3190657.

[106] Norman M Fraser and G Nigel Gilbert. “Simulating speech systems”. In: *Computer Speech & Language* 5.1 (1991), pp. 81–99. DOI: 10.1016/0885-2308(91)90019-M.

[107] Charles Carpenter Fries. *The structure of English*. New York: Hartcourt Brace, 1952[1887].

[108] Malte Gabsdil. “Clarification in spoken dialogue systems”. In: *Proceedings of the 2003 AAAI Spring Symposium. Workshop on Natural Language Generation in Spoken and Written Dialogue*. 2003, pp. 28–35.

[109] Massimo Cerruti Gaetano Berruto. *La Linguistica. Un Corso Introduttivo*. Ed. by UTET Università. 1st ed. Novara: De Agostini Scuola SpA, 2011.

[110] Vittorio Gallese and George Lakoff. "The brain's concepts: The role of the sensory-motor system in conceptual knowledge". In: *Cognitive neuropsychology* 22.3-4 (2005), pp. 455–479. DOI: 10.1080/02643290442000310.

[111] Neelansh Garg et al. "FlavorDB: a database of flavor molecules". In: *Nucleic Acids Research* 46.D1 (Oct. 2017), pp. D1210–D1216. DOI: 10.1093/nar/gkx957.

[112] Dirk Geeraerts and Hubert Cuyckens. *The Oxford Handbook of Cognitive Linguistics*. 1st ed. Oxford University Press, 2007.

[113] Danilo Giampiccolo et al. "The Third PASCAL Recognizing Textual Entailment Challenge". In: *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Association for Computational Linguistics, June 2007, pp. 1–9. URL: https://aclanthology.org/W07-1401/.

[114] Roberto Basile Giannini, Antonio Origlia, and Maria Di Maro. "Taking decisions in a Hybrid Conversational AI architecture using Influence Diagrams". In: (2024).

[115] Raymond W. Gibbs Jr. et al. "Taking a Stand on the Meanings of Stand: Bodily Experience as Motivation for Polysemy". In: *Journal of Semantics* 11.4 (Nov. 1994), pp. 231–251. DOI: 10.1093/jos/11.4.231.

[116] Arthur M Glenberg and Vittorio Gallese. "Action-based language: A theory of language acquisition, comprehension, and production". In: *cortex* 48.7 (2012), pp. 905–922.

[117] Adele Evabook Goldberg. *Constructions: A construction grammar approach to argument structure*. Cognitive Theory of Language Culture Series CTLC (CHUP). University of Chicago Press, 1995.

[118] Alvin I. Goldman. *Theory of human action*. Princeton Legacy Library. Princeton University Press, 2015[1977].

[119] Ariel Goldstein et al. "Thinking ahead: spontaneous prediction in context as a keystone of language in humans and machines". In: *bioRxiv* (2020), pp. 2020–12. DOI: 10.1101/2020.12.02.403477.

[120] Melvyn A Goodale and A David Milner. "Separate visual pathways for perception and action". In: *Trends in neurosciences* 15.1 (1992), pp. 20–25. DOI: 10.1016/0166-2236(92)90344-8.

[121] Noah D. Goodman and Michael C. Frank. "Pragmatic Language Interpretation as Probabilistic Inference". In: *Trends in Cognitive Sciences* 20.11 (2016), pp. 818–829. DOI: https://doi.org/10.1016/j.tics.2016.08.005.

[122] Andrew Gordon. "Commonsense interpretation of Triangle behavior". In: *Proceedings of the aaai conference on artificial intelligence*. AAAI'16. Phoenix, Arizona: AAAI Press, 2016, 3719–3725.

[123] Andrew Gordon and Jerry R Hobbs. "Coverage and competency in formal theories: A commonsense theory of memory". In: *Proceedings of the 2003 AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*. Mar. 2003, pp. 24–26.

[124] Paul Grice. *In the way of words*. Cambridge, Mass., 1989.

[125] Paul Grice H. "Logic and Conversation". In: *Syntax and Semantics*. Ed. by Peter Cole and Jerry L. Morgan. Vol. 3. New York: Academic Press, 1975, pp. 41–58.

[126] Ralph Grishman and Beth M Sundheim. "Message Understanding Conference-6: A Brief History". In: *The 16th International Conference on Computational Linguistics*. Vol. 1. COLING 1996. Copenhagen, Denmark: Association for Computational Linguistics, 1996, 466–471. DOI: 10.3115/992628.992709.

[127] Thomas R. Gruber. *A translation approach to portable ontology specifications*. 1993. DOI: https://doi.org/10.1006/knac.1993.1008.

[128] Nicola Guarino, Daniel Oberle, and Steffen Staab. "What is an ontology?" In: *Handbook on ontologies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1–17. DOI: 10.1007/978-3-540-92673-3_0.

[129] John J Gumperz. "The retrieval of socio-cultural knowledge in conversation". In: *Poetics Today* 1.1/2 (1979), pp. 273–286. DOI: 10.2307/1772050.

[130] David Gunning. "Machine Common Sense Concept Paper". In: *ArXiv* (2018). DOI: 10.48550/arXiv.1810.07528.

[131] John Haiman. "Dictionaries and encyclopedias". In: *Lingua* 50.4 (1980), pp. 329–357. DOI: https://doi.org/10.1016/0024-3841(80)90089-3.

[132] Kenneth R Hammond. *Human Judgment and Social Policy: Irreducible Uncertainty, Inevitable Error, Unavoidable Injustice*. Oxford University Press, Sept. 1996. DOI: 10.1093/oso/9780195097344.001.0001.

[133] Olaf Hauk, Ingrid Johnsrude, and Friedemann Pulvermüller. "Somatotopic Representation of Action Words in Human Motor and Premotor Cortex". In: *Neuron* 41.2 (2004), pp. 301–307. ISSN: 0896-6273. DOI: https://doi.org/10.1016/S0896-6273(03)00838-9.

[134] Shirley Anugrah Hayati et al. "Inspired: Toward sociable recommendation dialog systems". In: *ArXiv* (2020). DOI: 10.48550/arXiv.2009.14306.

[135] Patrick J Hayes. "The Naive Physics Manifesto". In: *Expert Systems in the Electronic Age*. Ed. by Donald Michie. Edinburgh, Scotland: Edinburgh University Press, 1979, pp. 242–270.

[136] James Henderson, Oliver Lemon, and Kallirroi Georgila. "Hybrid Reinforcement/Supervised Learning of Dialogue Policies from Fixed Data Sets". In: *Computational Linguistics* 34.4 (2008), pp. 487–511. DOI: 10.1162/coli.2008.07-028-R2-05-82.

[137] Jerry R Hobbs et al. "Interpretation as abduction". In: *Artificial intelligence* 63.1-2 (1993), pp. 69–142. DOI: 10.1016/0004-3702(93)90015-4.

[138] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[139] Aidan Hogan et al. "Knowledge Graphs". In: *ACM Comput. Surv.* 54.4 (July 2021). DOI: 10.1145/3447772.

[140] Frits L van Holthoon and David R Olson. *Common sense: The foundations for social science*. Vol. 6. Lanham: University Press of America, 1987.

[141] Paul Hoyningen-Huene. "Systematicity: The nature of science". In: *Philosophia* 36.2 (2008), pp. 167–180. DOI: 10.1007/s11406-007-9100-x.

[142] Xuming Hu et al. "Do Large Language Models Know about Facts?" In: *ArXiv* (2023). DOI: 10.48550/arXiv.2310.05177.

[143] Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. "Challenges in Building Intelligent Open-domain Dialog Systems". In: *ACM Transactions on Information Systems (TOIS)* 38.3 (2020), pp. 1–32. DOI: 10.1145/3383123.

[144] Yan Huang. *The Oxford Handbook of Pragmatics*. UK: Oxford University Press, 2017.

[145] Dell Hymes. "On communicative competence". In: *Sociolinguistics*. Ed. by J. Holmes J. Pride. Harmondsworth: Penguin Books, 1972, pp. 269–293.

[146] Filip Ilievski, Pedro Szekely, and Bin Zhang. "CSKG: The CommonSense Knowledge Graph". In: *The Semantic Web*. Ed. by Ruben Verborgh et al. Cham: Springer International Publishing, 2021, pp. 680–696. DOI: 10.1007/978-3-030-77385-4_41.

[147] Jeri J. Jaeger and John J. Ohala. "On the structure of phonetic categories". In: *Proceedings of the Tenth Annual Meeting of the Berkeley Linguistics Society*. Berkley Linguistic Society. Berkley, California, Feb. 1984, pp. 15–26. DOI: 10.3765/bls.v10i0.3313.

[148] Roman Jakobson. "Metalanguage as a linguistic problem". In: *Selected Writings VII. Contributions to Comparative Mythology. Studies in Linguistics and Philology,1972–1982.* Mouton Publishers, 1985[1976].

[149] Daniel Jannai et al. "Human or Not? A gamified approach to the Turing test". In: *ArXiv* (2023). DOI: 10.48550/arXiv.2305.20010.

[150] Julian Jara-Ettinger et al. "The naïve utility calculus: Computational principles underlying commonsense psychology". In: *Trends in cognitive sciences* 20.8 (2016), pp. 589–604.

[151] Finn V Jensen. "Bayesian networks basics". In: *AISB quarterly* (1996), pp. 9–22.

[152] Elisabetta Ježek. *Lessico: classi di parole, strutture, combinazioni*. Bologna: Il Mulino, 2005.

[153] Elisabetta Jezek and Rachele Sprugnoli. *Linguistica Computazionale. Introduzione all'analisi automatica dei testi*. Il Mulino, 2023.

[154] Shaoxiong Ji et al. "A survey on knowledge graphs: Representation, acquisition, and applications". In: *IEEE transactions on neural networks and learning systems* 33.2 (2021), pp. 494–514.

[155] Ziwei Ji et al. "Survey of Hallucination in Natural Language Generation". In: *ACM Computing Surveys* 55.12 (2023), pp. 1–38. DOI: 10.1145/3571730.

[156] Yiwei Jiang et al. “Recipe instruction semantics corpus (RISeC): Resolving semantic structure and zero anaphora in recipes”. In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. 2020, pp. 821–826. DOI: 10.18653/v1/2020.aacl-main.82.

[157] Yiwei Jiang et al. “CookDial: a dataset for task-oriented dialogs grounded in procedural documents”. In: *Applied Intelligence* 53.4 (2023), pp. 4748–4766. DOI: 10.1007/s10489-022-03692-0.

[158] Zhengbao Jiang et al. “How can we know what language models know?” In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 423–438.

[159] Pablo Jiménez, Javier Villalba Diez, and Joaquin Ordieres-Mere. “HOSHIN KANRI Visualization with Neo4j. Empowering Leaders to Operationalize Lean Structural Networks”. In: *Procedia CIRP* 55 (2016). 5th CIRP Global Web Conference - Research and Innovation for Future Production (CIRPe 2016), pp. 284–289. DOI: https://doi.org/10.1016/j.procir.2016.08.023.

[160] Mark Johnson. *The body in the mind: The bodily basis of meaning, imagination*. Chicago: University of Chicago Press, 1987.

[161] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd. Online manuscript released August 20, 2024. 2024. URL: https://web.stanford.edu/~jurafsky/slp3/.

[162] Jared Kaplan et al. “Scaling Laws for Neural Language Models”. In: *ArXiv* (2020). DOI: 10.48550/arXiv.2001.08361.

[163] Alex Kean and George Tsiknis. “Assumption-Based Reasoning and Clause Management Systems”. In: *Computational Intelligence* 8.1 (1992), pp. 1–24.

[164] Istvan Kecskes. “Encyclopaedic knowledge and cultural models”. In: *Cognitive pragmatics*. Ed. by Hans-Jörg Schmid. Berlin, Boston: Mouton De Gruyter, 2012. Chap. 7, pp. 175–198. DOI: 10.1515/9783110214215.175.

[165] Istvan Kecskes. *Intercultural Pragmatics*. New York: Oxford University Press, 2013.

[166] Ruth M Kempson. *Presupposition and the Delimitation of Semantics*. Vol. 15. New York: Cambridge University Press, 1975.

[167] Jacob Devlin Kenton, Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Vol. 1. Minneapolis, Minnesota. Association for Computational Linguistics, June 2019, p. 2. DOI: doi="10.18653/v1/N19-1423.

[168] Vid Kocijan et al. "The defeat of the Winograd Schema Challenge". In: *Artificial Intelligence* 325 (2023), p. 103971. DOI: https://doi.org/10.1016/j.artint.2023.103971.

[169] Stefan Kombrink et al. "Recurrent Neural Network Based Language Modeling in Meeting Recognition." In: *Proceedings of 12th Annual Conference of the International Speech Communication Association (Interspeech 2011)*. Vol. 11. Florence, Italy, Aug. 2011, pp. 2877–2880. DOI: 10.21437/Interspeech.2011-720.

[170] Yinghui Kong et al. "Bolt defect classification algorithm based on knowledge graph and feature fusion". In: *Energy Reports* 8 (2022). 2021 The 8th International Conference on Power and Energy Systems Engineering, pp. 856–863. DOI: https://doi.org/10.1016/j.egyr.2021.11.127.

[171] Sivarama Krishnan, R Guruvayur, and R Suchithra. "Design of A Machine Learning Model for Automatic Generation of Domain-Specific Ontologies". In: *Online International Interdisciplinary Research Journal* (Dec. 2018). ISSN: 2249-9598.

[172] Gerhard Lakemeyer and Bernhard Nebel. *Foundations of Knowledge Representation and Reasoning*. 1st. Berlin Heidelberg: Springer, 2005, pp. 1–12. DOI: 10.1007/3-540-58107-3.

[173] George Lakoff. "Cognitive semantics". In: *Meaning and Mental Representations*. Ed. by Umberto Eco, Marco Santambrogio, and Patrizia Violi. Indiana University Press, 1988, pp. 119–154.

[174] George Lakoff. "The invariance hypothesis: Is abstract reason based on image-schemas?" In: *Cognitive Linguistics* 1.1 (1990), pp. 39–74. DOI: doi:10.1515/cogl.1990.1.1.39.

[175] George Lakoff. *Women, fire, and dangerous things: What categories reveal about the mind*. 1st. University of Chicago Press, 2008.

[176] George Lakoff. *Moral politics: How liberals and conservatives think*. University of Chicago Press, 2016.

[177] Ronald W Langacker. *Foundations of cognitive grammar*. Vol. Volume I: Theoretical prerequisites. Stanford: Stanford University Press, 1987.

[178] Ronald W Langacker. "Nouns and verbs". In: *Language* 63.1 (1987), pp. 53–94. DOI: doi.org/10.2307/415384.

[179] Maarten Lemmens. "Cognitive semantics". In: *The Routledge handbook of semantics*. Ed. by Nick Riemer. 1st Edition. London: Routledge, 2015. Chap. 16, pp. 90–105. DOI: 10.4324/9781315685533.

[180] Noah Lemos. *Common sense: A contemporary defense*. New York: Cambridge University Press, 2004.

[181] Douglas B Lenat. "CYC: A large-scale investment in knowledge infrastructure". In: *Communications of the ACM* 38.11 (Nov. 1995), pp. 33–38. DOI: 10.1145/219717.219745.

[182] Douglas B Lenat and Ramanathan V. Guha. "The evolution of CycL, the Cyc representation language". In: *SIGART Bull.* 2.3 (June 1991), pp. 84–87. DOI: 10.1145/122296.122308.

[183] Hector Levesque, Ernest Davis, and Leora Morgenstern. "The winograd schema challenge". In: *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*. KR'12. Rome, Italy: AAAI Press, 2012, 552–561.

[184] Hector J. Levesque. "The Winograd Schema Challenge". In: *Logical Formalizations of Commonsense Reasoning*. AAI 2011 Spring Symposium (June 2011).

[185] Stephen C Levinson. *Pragmatics*. Cambridge: Cambridge University Press, 1983.

[186] Clive Staples Lewis. *Studies in words*. Cambridge University Press, 1990.

[187] Antonio Lieto et al. “The role of cognitive architectures in general artificial intelligence”. In: *Cognitive Systems Research* 48 (2018). Cognitive Architectures for Artificial Minds, pp. 1–3. DOI: 0.1016/j.cogsys.2017.08.003.

[188] Stephanie Lin, Jacob Hilton, and Owain Evans. “TruthfulQA: Measuring How Models Mimic Human Falsehoods”. In: *ArXiv* (2021). DOI: 10.48550/arXiv.2109.07958.

[189] Dennis V Lindley. *Understanding uncertainty*. John Wiley & Sons, 2013.

[190] Hugo Liu and Push Singh. “ConceptNet—a practical commonsense reasoning tool-kit”. In: *BT Technology Journal* 22.4 (2004), pp. 211–226. DOI: 10.1023/B:BTTJ.0000047600.45421.6d.

[191] Pengfei Liu et al. “Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing”. In: *ACM Comput. Surv.* 55.9 (Jan. 2023), pp. 1–35. DOI: 10.1145/3560815.

[192] V.S.M.R. Lo Cascio. “Semantica lessicale e i criteri di collocazione nei dizionari bilingui a stampa ed elettronici”. In: *Lessico e grammatica: teorie linguistiche e applicazioni lessicografiche*. Ed. by V.S.M.R. Lo Cascio and T. De Mauro. Roma: Bulzoni, 1997, pp. 63–87.

[193] George F. Luger. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. 6th. Pearson - Addison Wesley, 2016.

[194] Yi Ma et al. “Knowledge graph inference for spoken dialog systems”. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 5346–5350. DOI: 10.1109/ICASSP.2015.7178992.

[195] Brian MacWhinney and William O’Grady. *The Handbook of Language Emergence*. John Wiley & Sons, 2015. DOI: 10.1002/9781118346136.

[196] Gary Marcus and Ernest Davis. *Rebooting AI: Building Artificial Intelligence We Can Trust*. Vintage, 2019.

[197] Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. “Building a Large Annotated Corpus of English: The Penn Treebank”. In: *Computational Linguistics* 19.2 (1993). Ed. by Julia Hirschberg, pp. 313–330.

[198] Javier Marın et al. “Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.1 (2021), pp. 187–203.

[199] Naser Masri et al. "Knowledge-Based Systems Survey". In: *International Journal of Academic Engineering Research (IJAER)* 3.7 (2019), pp. 1–22. ISSN: 2643-9085.

[200] John McCarthy. *Programs with Common Sense*. Stanford, CA, 1959.

[201] Ken McRae et al. "Semantic feature production norms for a large set of living and nonliving things". In: *Behavior Research Methods* 37.4 (2005), pp. 547–559. DOI: 10.3758/BF03192726.

[202] Jim Melton. "SQL language summary". In: *ACM Comput. Surv.* 28.1 (Mar. 1996), 141–143. DOI: 10.1145/234313.234374.

[203] Sabrina Mennella. "Semantic Annotation for Extracting Commonsense Knowledge Information Structure". In: *Il dialogo tra scienze linguistiche e nuove tecnologie* (June 2024), p. 41.

[204] Sabrina Mennella, Maria Di Maro, and Martina Di Bratto. "Common Sense Knowledge graph generation for information-gap requests in dialogue systems". In: *Proceedings of the 16th International Cognitive Linguistics Conference (ICLC16)*. Düsseldorf, Germany, 2023.

[205] Sabrina Mennella, Maria Di Maro, and Martina Di Bratto. "Estimating Commonsense Knowledge from a Linguistic Analysis on Information Distribution". In: *Proceedings of the Sixth International Conference on Computational Linguistics in Bulgaria (CLIB 2024)*. 2024, pp. 257–263.

[206] Tomas Mikolov et al. "Recurrent neural network based language model." In: *Proc. Interspeech 2011*. Vol. 2. 3. Makuhari. Chiba, Japan, Sept. 2010, pp. 1045–1048. DOI: 10.21437/Interspeech.2010-343.

[207] Tomás Mikolov et al. "Efficient Estimation of Word Representations in Vector Space". In: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2013.

[208] George A Miller. "WordNet: a lexical database for English". In: *Commun. ACM* 38.11 (Nov. 1995), pp. 39–41. DOI: 10.1145/219717.219748.

[209] Justin J Miller. "Graph database applications and concepts with Neo4j". In: *Proceedings of the southern association for information systems conference*. Vol. 2324. 36. Atlanta, GA, USA, 2013, pp. 141–147.

[210] Marvin Minsky. *Society of mind*. Simon and Schuster, 1988.

[211] Marvin Minsky. "Commonsense-based interfaces". In: *Commun. ACM* 43.8 (Aug. 2000), pp. 66–73. DOI: 10.1145/345124.345145.

[212] Marvin Minsky et al. *A framework for representing knowledge*. 1974.

[213] George Edward Moore. "A Defence of Common Sense". In: *Contemporary British Philosophy, Second Series*. Ed. by J. H. Muirhead. George Allen and Unwin, 1925.

[214] L. Morgenstern and C Ortiz. "The Winograd schema challenge: Evaluating Progress in Commonsense Reasoning". In: *Proceedings of the Twenty-Seventh Conference on Innovative Applications of Artificial Intelligence (AAAI-15)*. Palo Alto, California USA, 2015, pp. 4024–4025.

[215] Leora Morgenstern. "Mid-sized axiomatizations of commonsense problems: A case study in egg cracking". In: *Studia Logica* 67 (2001), pp. 333–384. DOI: 10.1023/A:1010512415344.

[216] Leora Morgenstern. "Knowledge Representation and Commonsense Reasoning: Reviews of Four Books". In: *Artificial Intelligence* 170.18 (2006), pp. 1239–1250. DOI: 10.1016/j.artint.2006.10.012.

[217] Leora Morgenstern. *Pronoun Disambiguation Problems*. 2016. URL: https://commonsensereasoning.org/disambiguation.html.

[218] Nasrin Mostafazadeh et al. "A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kevin Knight, Ani Nenkova, and Owen Rambow. San Diego, California: Association for Computational Linguistics, June 2016, pp. 839–849. DOI: 10.18653/v1/N16-1098.

[219] Erik T Mueller. *Commonsense Reasoning: An Event Calculus-Based Approach*. 2nd Edition. Morgan Kaufmann, 2014.

[220] Allen Newell, John C Shaw, and Herbert A Simon. "Report on a general problem solving program". In: *IFIP Congress*. Vol. 256. Pittsburgh, PA. 1959, p. 64.

[221] Tuan-Phong Nguyen et al. “Refined commonsense knowledge from large-scale web contents”. In: *IEEE Transactions on Knowledge and Data Engineering* (2022). DOI: 10.48550/arXiv.2112.04596.

[222] Ikujiro Nonaka. “The knowledge-creating company”. In: *The Economic Impact of Knowledge*. Ed. by Tony Siesfeld, Jacquelyn Cefola, and Dale Neef. London: Routledge, 2009 [1998]. Chap. 13, pp. 175–187. DOI: 10.4324/9780080505022.

[223] Natalya F Noy et al. “Creating Semantic Web contents with Protege-2000”. In: *IEEE Intelligent Systems* 16.2 (2001), pp. 60–71. DOI: 10.1109/5254.920601.

[224] Rafael E Núñez, Rafael Núñez, and Walter J Freeman. *Reclaiming cognition: The primacy of action, intention and emotion*. Imprint Academic, 1999.

[225] Charles Kay Ogden and Ivor Armstrong Richards. “The meaning of meaning: A study of the influence of thought and of the science of symbolism”. In: (1923).

[226] Antonio Origlia and Maria Di Maro. “A Linguistically Motivated Approach to Hybrid Conversational AI with the FANTASIA Plugin”. In: *Proceedings of the 24th ACM International Conference on Intelligent Virtual Agents*. 2024, pp. 1–3.

[227] Antonio Origlia et al. “FANTASIA: a framework for advanced natural tools and applications in social, interactive approaches”. In: *Multimedia Tools and Applications* 78 (2019), pp. 13613–13648. DOI: 10.1007/s11042-019-7362-5.

[228] Antonio Origlia et al. “A multi-source graph representation of the movie domain for recommendation dialogues analysis”. In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. June 2022, pp. 1297–1306.

[229] Antonio Origlia et al. “Developing Embodied Conversational Agents in the Unreal Engine: The FANTASIA Plugin”. In: *Proceedings of the 30th ACM International Conference on Multimedia*. Lisboa, Portugal, 2022, pp. 6950–6951. DOI: 10.1145/3503161.3550065.

[230] Antonio Origlia et al. “Developing embodied conversational agents in the unreal engine: the FANTASIA Plugin”. In: *Proceedings of the 30th ACM International Conference on Multimedia*. 2022, pp. 6950–6951.

[231] Inès Osman, Sadok Ben Yahia, and Gayo Diallo. “Ontology Integration: Approaches and Challenging Issues”. In: *Information Fusion* 71 (2021), pp. 38–63. DOI: https://doi.org/10.1016/j.inffus.2021.01.007.

[232] Steve Oswald. “Pragmatics for argumentation”. In: *Journal of Pragmatics* 203 (2023), pp. 144–156. DOI: 10.1016/j.pragma.2022.12.001.

[233] Mostafa Oualif. “Presupposition: A Semantic or Pragmatic Phenomenon?” In: *Arab World English Journal (AWEJ)* 8.3 (2017), pp. 46–59. DOI: 10.2139/ssrn.3053527.

[234] Long Ouyang et al. “Training language models to follow instructions with human feedback”. In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh et al. Vol. 35. 2022, pp. 27730–27744.

[235] Xiaoman Pan et al. “Knowledge-in-context: Towards knowledgeable semi-parametric language models”. In: *ArXiv* (2022). DOI: 10.48550/arXiv.2210.16433.

[236] Joon Sung Park et al. “Generative Agents: Interactive Simulacra of Human Behavior”. In: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. UIST ’23. San Francisco, CA, USA: Association for Computing Machinery, 2023, pp. 1–22. DOI: 10.1145/3586183.3606763.

[237] Rebecca J. Passonneau. “Computing reliability for coreference annotation”. In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC‘04)*. Ed. by Maria Teresa Lino et al. Lisbon, Portugal: European Language Resources Association (ELRA), May 2004.

[238] Rebecca J. Passonneau. “Measuring agreement on set-valued items (MASI) for semantic and pragmatic annotation”. In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC‘06)*. Genoa, Italy: European Language Resources Association (ELRA), 2006.

[239] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. 340 Pine Street, Sixth Floor, San Francisco, CA, United States: Morgan Kaufmann Publishers Inc., 1988.

[240] Charles S. Peirce. "A theory of probable inference." In: *Studies in logic by members of the Johns Hopkins University*. Ed. by Charles S. Peirce. Little, Brown and Co, 1883.

[241] Baolin Peng et al. "Composite Task-Completion Dialogue Policy Learning via Hierarchical Deep Reinforcement Learning". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017. DOI: 10.18653/v1/D17-1237.

[242] Ciyuan Peng et al. "Knowledge graphs: Opportunities and challenges". In: *Artificial Intelligence Review* 56.11 (2023), pp. 13071–13102. DOI: 10.1007/s10462-023-10465-9.

[243] Matthew E Peters et al. "Semi-supervised sequence tagging with bidirectional language models". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1756–1765. DOI: 10.18653/v1/P17-1161.

[244] Fabio Petroni et al. "Language Models as Knowledge Bases?" 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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2463–2473. DOI: 10.18653/v1/D19-1250.

[245] Miriam R.L. Petruck. "Frame semantics". In: *Handbook of Pragmatics Online* 2 (1996[2022]). DOI: 10.1075/hop.2.fra1.

[246] Martha E. Pollack. "The Uses of Plans". In: *Artificial Intelligence* 57.1 (1992), pp. 43–68. DOI: 10.1016/0004-3702(92)90104-6.

[247] Leon Pompa. *Vico: A study of the'New Science'*. Cambridge University Press, 1990.

[248] Simone Paolo Ponzetto, Michael Strube, et al. “Deriving a large scale taxonomy from Wikipedia”. In: *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 2*. AAAI’07. Vancouver, British Columbia, Canada: AAAI Press, 2007, 1440–1445.

[249] Libo Qin et al. “A Survey on Spoken Language Understanding: Recent Advances and New Frontiers”. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence - Survey Track*. IJCAI-21. International Joint Conferences on Artificial Intelligence, 2021, pp. 4577–4584. DOI: 10.24963/ijcai.2021/622.

[250] Alec Radford et al. “Language models are unsupervised multitask learners”. In: *OpenAI blog* 1.8 (2019), p. 9.

[251] Hannah Rashkin et al. “Event2Mind: Commonsense Inference on Events, Intents, and Reactions”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 463–473. DOI: 10.18653/v1/P18-1043.

[252] Benjamin W. Redekop. “Thomas Reid and the problem of induction: from common experience to common sense”. In: *Studies in History and Philosophy of Science Part A* 33.1 (2002), pp. 35–57. DOI: 10.1016/S0039-3681(01)00022-X.

[253] Adam Roberts, Colin Raffel, and Noam Shazeer. “How Much Knowledge Can You Pack Into the Parameters of a Language Model?” In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Nov. 2020, pp. 5418–5426. DOI: 10.18653/v1/2020.emnlp-main.437.

[254] Ian Robinson, Jim Webber, and Emil Eifrem. *Graph Databases: New Opportunities for Connected Data*. 2nd. O’Reilly Media, Inc., 2015.

[255] Marko A Rodriguez and Peter Neubauer. “Constructions from dots and lines”. In: *ArXiv* (2010). DOI: 10.48550/arXiv.1006.2361.

[256] Eleanor H. Rosch. “On the internal structure of perceptual and semantic categories”. In: *Cognitive Development and Acquisition of Language*. Ed. by Timothy E. Moore. San Diego: Academic Press, 1973, pp. 111–144.

[257] Sophia Rosenfeld. *Common sense: A political history*. Harvard University Press, 2011.

[258] Rema Rossini Favretti. *Frames, Corpora and Knowledge Representation*. Bologna: Bononia University Press, 2008.

[259] Timothy N Rubin et al. "Decoding brain activity using a large-scale probabilistic functional-anatomical atlas of human cognition". In: *PLoS Computational Biology* 13.10 (2017), e1005649. DOI: 10.1371/journal.pcbi.1005649.

[260] T. Ruffman, L. Slade, and M. Taumoepeau. "Theory of mind and language ability: Understanding the bigger picture". In: *Invited paper, Web conference on co-evolution of language and theory of mind. Retrieved August*. Vol. 6. 21. 2004, p. 2008.

[261] Josef Ruppenhofer et al. *FrameNet II: Extended theory and practice*. Tech. rep. International Computer Science Institute, 2016.

[262] Paul Rusnock and Rolf George. *De la méthode mathématique et correspondance avec Exner*. Trans. by Paul Rusnock and Rolf George. Introduction par C. Maigné et J. Šebestík. Paris: Vrin, 2008.

[263] Bertrand Russell. "On denoting". In: *Mind* 14.56 (1905), pp. 479–493.

[264] Walid S. Saba. "Language, logic and ontology: Uncovering the structure of commonsense knowledge". In: *International Journal of Human-Computer Studies* 65.7 (2007). Knowledge representation with ontologies: Present challenges - Future possibilities, pp. 610–623. DOI: https://doi.org/10.1016/j.ijhcs.2007.02.002.

[265] Walid S Saba. "Commonsense Knowledge, Ontology and Ordinary Language". In: *International Journal of Reasoning-based Intelligent Systems* 2.1 (2010), pp. 36–50. DOI: doi.org/10.1504/IJRIS.2010.029813.

[266] Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. "A simplest systematics for the organization of turn-taking for conversation". In: *language* 50.4 (1974), pp. 696–735.

[267] Hiroaki Sakoe and Seibi Chiba. "Dynamic programming algorithm optimization for spoken word recognition". In: *IEEE transactions on acoustics, speech, and signal processing* 26.1 (1978), pp. 43–49.

[268] Alireza Salemi et al. “LaMP: When Large Language Models Meet Personalization”. In: *ArXiv* (2023). DOI: 10.48550/arXiv.2304.11406.

[269] Erik Sandewall. *Features and Fluents: The representation of knowledge about dynamical systems*. Oxford university press, 1995.

[270] V. Sanh. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *ArXiv* (2019). DOI: 10.48550/arXiv.1910.01108.

[271] Maarten Sap et al. “ATOMIC: an atlas of machine commonsense for if-then reasoning”. In: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*. Vol. 33. AAAI’19/IAAI’19/EAAI’19. AAAI Press, 2019, pp. 3027–3035. DOI: 10.1609/aaai.v33i01.33013027.

[272] Maarten Sap et al. “Commonsense reasoning for natural language processing”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Online: Association for Computational Linguistics, July 2020, pp. 27–33. DOI: 10.18653/v1/2020.acl-tutorials.7. URL: https://www.youtube.com/watch?v=InIffoMnV7k.

[273] Maarten Sap et al. “Neural Theory-of-Mind? On the Limits of Social Intelligence in Large LMs”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 3762–3780. DOI: 10.18653/v1/2022.emnlp-main.248.

[274] Alexandre Schiele and Bernard Schiele. “Self-awareness and common sense– The paradox of AI: A dispassionate look”. In: *AI and Common Sense*. Ed. by Bernard Schiele Martin W. Bauer. London, UK: Routledge, 2024, pp. 30–44.

[275] Bernard Schiele and Martin W. Bauer. “AI goes to the movies: Fast, intermediate and slow common sense”. In: *AI and Common Sense*. London,UK: Routledge, 2024, pp. 241–252.

[276] Karin Kipper Schuler and Martha S. Palmer. “Verbnet: a broad-coverage, comprehensive verb lexicon”. AAI3179808. PhD thesis. USA: University of Pennsylvania, 2005.

[277] John R. Searle. *Minds, Brains, and Programs*. Vol. 3. 3. Behavioral and Brain Science, 1980, pp. 417–457.

[278] Claude Elwood Shannon. *The Mathematical Theory of Communication*. Vol. 27. 3. Nokia Bell Labs, 1949, pp. 379–423.

[279] Edward Shortliffe. *Computer-Based Medical Consultations: MYCIN*. 1st Ed. Elsevier, 2012.

[280] Push Singh, Marvin Minsky, and Ian Eslick. "Computing commonsense". In: *BT Technology Journal* 22.4 (2004), pp. 201–210.

[281] Push Singh et al. "Open Mind Common Sense: Knowledge Acquisition from the General Public". In: *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 1223–1237.

[282] Amit Singhal. "Introducing the Knowledge Graph: things, not strings". In: *Official Google Blog* 5.16 (2012), p. 3.

[283] Ramakrishnan Sivakumar and P.V. Arivoli. "Ontology Visualization PROTÉGÉ Tools– A Review". In: *International Journal of Advanced Information Technology (IJAIT)* 1.4 (Aug. 2011).

[284] Robyn Speer, Joshua Chin, and Catherine Havasi. "Conceptnet 5.5: An open multilingual graph of general knowledge". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. Vol. 31. AAAI'17 1. San Francisco, California, USA: AAAI Press, 2017, 4444—4451.

[285] Dan Sperber. *Relevance: Communication and cognition*. 2nd. Oxford UK and Cambridge USA: Blackwell, 1986 [1995].

[286] R Stalnaker. "Pragmatic presuppositions". In: *Semantics and Philosophy*. Ed. by M. Munitz and P. Unger. New York: University Press, 1974, 197–214.

[287] Robert Stalnaker. "Common ground". In: *Linguistics and Philosophy* 25.5/6 (Dec. 2002), pp. 701–721.

[288] Todd Andrew Stephenson. "An introduction to Bayesian network theory and usage". In: *Idiap Research Report* (2000).

[289] Shane Storks, Qiaozi Gao, and Joyce Y Chai. "Commonsense reasoning for natural language understanding: A survey of benchmarks, resources, and approaches". In: *ArXiv* (Apr. 2019), pp. 1–60. DOI: 10.48550/arXiv.1904.01172.

[290] Shane Storks, Qiaozi Gao, and Joyce Y Chai. "Recent Advances in Natural Language Inference: A Survey of Benchmarks, Resources, and Approaches". In: *ArXiv* (Feb. 2020). DOI: 10.48550/arXiv.1904.01172.

[291] Peter F Strawson. "On referring". In: *Mind* 59.235 (1950), pp. 320–344.

[292] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. "Yago: a core of semantic knowledge". In: *Proceedings of the 16th international conference on World Wide Web*. WWW '07. Banff, Alberta, Canada: Association for Computing Machinery, 2007, pp. 697–706. DOI: 10.1145/1242572.1242667.

[293] Fabian M Suchanek and Gerhard Weikum. "Knowledge bases in the age of big data analytics". In: *Proc. VLDB Endow.* 7.13 (Aug. 2014), pp. 1713–1714. DOI: 10.14778/2733004.2733069.

[294] Ahmet Süerdem. "The challenges and opportunities in large language models: Navigating the perils of stochastic and scholastic parrots in artificial understanding and common sense". In: ed. by Bernard Schiele Martin W. Bauer. London: Routledge, 2024. Chap. 13, pp. 195–212.

[295] Karen Sullivan. "Three levels of framing". In: *Wiley Interdisciplinary Reviews: Cognitive Science* 14.5 (2023), e1651. DOI: 10.1002/wcs.1651..

[296] Adrian Sureshkumar. *Ansprolog* programming environment (ape): Investigating software tools for answer set programming through the implementation of an integrated development environment*. 2006.

[297] Alon Talmor et al. "oLMpics-On What Language Model Pre-training Captures". In: *Transactions of the Association for Computational Linguistics* 8 (Dec. 2020), pp. 743–758. DOI: 10.1162/tacl_a_00342.

[298] Niket Tandon, Gerard De Melo, and Gerhard Weikum. "WebChild 2.0: Fine-Grained Commonsense Knowledge Distillation". In: *Proceedings of ACL 2017, System Demonstrations*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 115–120.

[299] Niket Tandon et al. "Webchild: Harvesting and organizing commonsense knowledge from the web". In: *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. WSDM '14. New York, New York, USA: Association for Computing Machinery, 2014, pp. 523–532. DOI: 10.1145/2556195.2556245.

[300] John R. Taylor. *Linguistic categorization*. Oxford University Press, 2003.

[301] Joshua B. Tenenbaum et al. "How to Grow a Mind: Statistics, Structure, and Abstraction". In: *Science* 331.6022 (2011), pp. 1279–1285. DOI: 10.1126/science.1192788.

[302] Maxim Tkachenko et al. *Label Studio: Data labeling software*. Open source software available from https://github.com/heartexlabs/label-studio. 2020-2022. URL: https://github.com/heartexlabs/label-studio.

[303] Michael Tomasello. "Learning through others". In: *Daedalus* 133.1 (2004), pp. 51–58.

[304] Michael Tomasello. *Origins of human communication*. MIT press, 2010.

[305] Alberto Tonon et al. "Contextualized ranking of entity types based on knowledge graphs". In: *Journal of Web Semantics* 37-38 (2016), pp. 170–183. DOI: 10.1016/j.websem.2015.12.005.

[306] Hugo Touvron et al. "LLaMA: Open and Efficient Foundation Language Models". In: *ArXiv* (2023). DOI: 10.48550/arXiv.2302.13971.

[307] David R Traum and Staffan Larsson. "The information state approach to dialogue management". In: *Current and New Directions in Discourse and Dialogue* (2003), pp. 325–353. DOI: 10.1007/978-94-010-0019-2_15.

[308] Giacomo Turbanti. *Philosophy of Communication*. London: Palgrave Macmillan, 2022. DOI: 10.1007/978-3-031-12463-1.

[309] Alan M Turing. "Computing Machinery and Intelligence". In: *Mind* 49.36 (1950), pp. 433–460.

[310] Teun A. Van Dijk. "The Discourse-Knowledge Interface". In: *Critical Discourse Analysis: Theory and Interdisciplinarity*. Ed. by Gilbert Weiss and Ruth Wodak. London: Palgrave Macmillan UK, 2003, pp. 85–109. DOI: 10.1057/9780230514560_5.

[311] A Vaswani et al. “Attention Is All You Need”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Long Beach, California, USA: Curran Associates Inc., 2017, 6000–6010.

[312] Niki Verschueren, Walter Schaeken, and Géry d’Ydewalle. “A dual-process specification of causal conditional reasoning”. In: *Thinking & Reasoning* 11.3 (2005), pp. 239–278. DOI: 10.1080/13546780442000178.

[313] Max Völkel et al. “Semantic Wikipedia”. In: *Proceedings of the 15th International Conference on World Wide Web*. WWW ’06. Edinburgh, Scotland: Association for Computing Machinery, 2006, pp. 585–594.

[314] Bernhard Waldenfels and J Claude Evans. “The Despised Doxa* Husserl and the Continuing Crisis of Western Reason”. In: *Research in Phenomenology* 12 (1982), pp. 21–38. DOI: 10.1163/156916482x00035.

[315] Jimmy Wales. “Wikipedia in the free culture revolution”. In: *Companion to the 20th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*. OOPSLA ’05. San Diego, CA, USA: Association for Computing Machinery, 2005, pp. 5–5. DOI: 10.1145/1094855.1094859.

[316] Warren E Walker et al. “Defining uncertainty: a conceptual basis for uncertainty management in model-based decision support”. In: *Integrated assessment* 4.1 (2003), pp. 5–17.

[317] Richard S Wallace. “The Anatomy of A.L.I.C.E.” In: *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*. Ed. by Robert Epstein, Gary Roberts, and Grace Beber. Dordrecht: Springer Netherlands, 2009. DOI: 10.1007/978-1-4020-6710-5_13.

[318] Alex Wang et al. “SuperGLUE: a stickier benchmark for general-purpose language understanding systems”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. (NeurIPS 2019). 2019.

[319] Dingmin Wang et al. “Fast and Scalable Dialogue State Tracking with Explicit Modular Decomposition”. In: *ArXiv* (2021). DOI: 10.48550/arXiv.2004.10663.

[320] Hongru Wang et al. “A survey of the evolution of language model-based dialogue systems”. In: *arXiv preprint arXiv:2311.16789* (2023).

[321] Xuena Wang et al. “Emotional intelligence of large language models”. In: *Journal of Pacific Rim Psychology* 17 (2023). DOI: 10.1177/18344909231213958.

[322] Duncan J Watts. “Common Sense and Sociological Explanations”. In: *American Journal of Sociology* 120.2 (2014), pp. 313–351.

[323] Jim Webber. “A programmatic introduction to Neo4j”. In: *Proceedings of the 3rd Annual Conference on Systems, Programming, and Applications: Software for Humanity*. SPLASH ’12. Tucson, Arizona, USA: Association for Computing Machinery, 2012, pp. 217–218. DOI: 10.1145/2384716.2384777.

[324] Mark E. Whiting and Duncan J. Watts. “A framework for quantifying individual and collective common sense”. In: *Proceedings of the National Academy of Sciences* 121.4 (2024), e2309535121. DOI: 10.1073/pnas.2309535121.

[325] Terry Winograd. “Understanding natural language”. In: *Cognitive Psychology* 3.1 (1972), pp. 1–191. DOI: https://doi.org/10.1016/0010-0285(72)90002-3.

[326] Lewis Wolpert. *The unnatural nature of science*. Cambridge, Mass.: Harvard University Press, 1994.

[327] Rongwu Xu et al. “The Earth is Flat because...: Investigating LLMs’ Belief towards Misinformation via Persuasive Conversation”. In: *ArXiv* (2023). DOI: 10.48550/arXiv.2312.09085.

[328] Da Yin et al. “A survey of knowledge-intensive nlp with pre-trained language models”. In: *ArXiv* (2022). DOI: 10.48550/arXiv.2202.08772.

[329] Jiahao Ying et al. “Intuitive or Dependent? Investigating LLMs’ Robustness to Conflicting Prompts”. In: *ArXiv* (2023). DOI: 10.48550/arXiv.2309.17415.

[330] Liang-Jun Zang et al. “A survey of commonsense knowledge acquisition”. In: *Journal of Computer Science and Technology* 28.4 (2013), pp. 689–719. DOI: 10.1007/s11390-013-1369-6.

[331] Xiaoying Zhang et al. "SGP-TOD: Building Task Bots Effortlessly via Schema-Guided LLM Prompting". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Singapore: Association for Computational Linguistics, 2023, pp. 13348–13369. DOI: 10.18653/v1/2023.findings-emnlp.891.

[332] Zheng Zhang et al. "Recent advances and challenges in task-oriented dialog systems". In: *Science China Technological Sciences* 63.10 (2020), pp. 2011–2027. DOI: 10.1007/s11431-020-1692-3.

[333] Wayne Xin Zhao et al. "A survey of large language models". In: *ArXiv* (2023). DOI: 10.48550/arXiv.2303.18223.

[334] Shen Zheng, Jie Huang, and Kevin Chen-Chuan Chang. "Why Does ChatGPT Fall Short in Providing Truthful Answers?" In: *ArXiv* (2023). DOI: 10.48550/arXiv.2304.10513.

[335] Ce Zhou et al. "A comprehensive survey on pretrained foundation models: A history from bert to chatgpt". In: *ArXiv* (2023). DOI: 10.48550/arXiv.2302.09419.

[336] Hao Zhou et al. "Commonsense knowledge aware conversation generation with graph attention." In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. IJCAI'18. Stockholm, Sweden: AAAI Press, 2018, pp. 4623–4629.

[337] Ming Zhou et al. "Progress in Neural NLP: Modeling, Learning, and Reasoning". In: *Engineering* 6.3 (2020), pp. 275–290. DOI: https://doi.org/10.1016/j.eng.2019.12.014.

[338] Qi Zhu et al. "ConvLab-2: An Open-Source Toolkit for Building, Evaluating, and Diagnosing Dialogue Systems". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, July 2020. DOI: 10.18653/v1/2020.acl-demos.19.

APPENDIX

A

FRAMES DESCRIPTION

- **Absorb_heat.** An **Entity** (generally food) is exposed to a **Heat_source** whose **Temperature** may also be specified. Generally, the Entity undergoes some sort of change as a result of this process.

  Frame Elements:

  - Entity: The Entity (often a food item) that absorbs heat
  - Manner: Any description of the cooking event which is not covered by more specific FEs, including secondary effects (quietly, loudly), and general descriptions comparing events (the same way).
  - Container: The Container that holds the Entity that absorbs the heat.
  - Heat_source: This FE identifies the object that directly supplies heat to the Entity.
  - Duration: The length of time that the Entity is exposed to heat.
  - Medium: Medium is a substance through which heat is absorbed by the Entity.
  - Temperature: The temperature that the Entity stays at, resulting in change.
  - Manner: Any description of the cooking event which is not covered by more specific FEs, including secondary effects (quietly, loudly), and general descriptions comparing events (the same way).

- **Apply_heat.** A **Cook** applies heat to food, where the **Temperature_setting** of the heat and **Duration** of application may be specified. A **Heating_instrument**, generally indicated by a locative phrase, may also be expressed. Some cooking

methods involve the use of a **Medium** (e.g., milk or water) by which heat is transferred to the food.

Frame Elements:

- Temperature_setting: This FE identifies the Temperature_setting of the Heating_instrument for the Food.
- Heating_instrument: This FE identifes the entity that directly supplies heat to the Food.
- Food: Food is the entity to which heat is applied by the Cook
- Container: The Container holds the Food to which heat is applied.
- Duration: Duration is the amount of Time heat is applied to the Food.
- Medium: Medium is the substance through which heat is applied to the Food.
- Place: This FE identifies the Place where the heat application occurs

- **Cause_change_of_phase.** A **Cause** or **Agent** causes a **Patient** to undergo a change of phase. The **Result** of the change may be given, along with the **Initial_state** and the **Circumstances** under which the change can occur. This frame describes causation of a change of a Patient between different phases (e.g., solid to liquid or frozen to "unfrozen").

  Frame Elements:

  - Patient: The Patient undergoes a change of phase brought about by the Agent.
  - Container: This FE identifies the Container that holds the Patient whose phase is being changed .
  - Degree: Degree to which the Agent changes the phase of the Patient
  - Instrument: This FE identifies the Instrument with which the Agent intentionally affects the Patient.
  - Result: Result is the result of the change of phase.
  - Time: This FE identifies the Time at which the change of phase occurs.

- **Cause_motion.** An **Agent** causes a **Theme** to move from a **Source**, along a **Path**, to a **Goal**. Different members of the frame emphasize the trajectory to

different degrees, and a given instance of the frame will usually leave some of the **Source**, **Path**, and/or **Goal** implicit.

Frame Elements:

- Theme: The FE Theme is generally an NP Object.
- Goal: The FE Goal is the point at which the Theme ends up as a result of the motion.
- Source: The FE Source is the starting point of motion.

- **Cause_temperature_change.** In this frame, an **Agent** changes the temperature of an **Item**. A **Temperature_goal** can specify the desired temperature. A **Temperature_change** can also be indicated. The **Temperature_start** indicates the initial temperature.

  Frame Elements:

  - Item: The Item undergoes the temperature change.
  - Container: This FE identifies the Container that holds the Item whose temperature is being changed.
  - Time: This FE identifies the Time when the Agent changes the temperature of the Item.
  - Temperature_goal: The Temperature_goal is the temperature to which the Agent is heating or cooling the Item.

- **Cause_to_amalgamate.** These words refer to an **Agent** joining **Parts** to form a **Whole**. The **Parts** may also be encoded as **Part_1** and **Part_2**. There is a symmetrical relationship between the components that undergo the process, and afterwards the **Parts** are consumed and are no longer distinct entities.

  Frame Elements:

  - Parts: This FE identifies the Parts entities being combined, and is often expressed in a single plural NP, usually the direct object of a verb.
  - Whole: This FE identifies Whole as the entity resulting from combination of Parts. When overtly expressed, it is usually a PP Complement (often headed by into)
  - Means: This FE identifies the Means by which the amalgamating occurs.
  - Place: The location that the event occurs

  - Time: The time at which the amalgamation occurs.
  - Result: This FE is used for indicating the Result of amalgamating.

- **Cause_to_be_dry.** An **Agent** causes a **Dryee** (either a surface or an entire entity, inside and out) to become dry. This should not include examples like "drying tears" or "drying spills," as these are in the **Removing** frame.
  Frame Elements:
  - Dryee: The Dryee is the entity which has the water removed from it.
  - Place: This FE identifies the Place where the Agent intentionally affects the Dryee.
  - Temperature: This FE identifies the Temperature at which heat is applied to the Dryee.
  - Duration: This FE identifies the length of Time that the drying takes.

- **Cause_to_be_included.** An **Agent** or **Cause** makes a **New_member** part of a **Group**. The **Group** may be represented by an individual **Existing_member** if it implies the existence of a set of members.
  Frame Elements:
  - Existing_member: Something that is already a member of the Group and represents the Group as a whole.
  - Place: The location in which a New_member is entered into a pre-existing Group.
  - New_member: An item that becomes a part of the Group.
  - Group: The Group is anything that can be conceptualized as a complex collection of parts or ingredients, including through metonymy like containers or locations, which New_members can join.

- **Cause_to_continue.** An **Agent** or **Cause** causes a **Process** or **State** to continue.
  Frame Elements:
  - State: The State that is maintained by the Agent or Cause.
  - Process: A Process that is maintained by the Agent or Cause
  - Place: The location at which the Process continues.

  - Duration: The length of time for which the Agent maintain the State or Process.

- **Cause_to_fragment.** An **Agent** suddenly and often violently separates the **Whole_patient** into two or more smaller **Pieces**, resulting in the **Whole_patient** no longer existing as such. Several lexical items are marked with the semantic type Negative, which indicates that the fragmentation is necessarily judged as injurious to the original **Whole_patient**.

  Frame Elements:

  - Whole_Patient: The entity which is destroyed by the Agent and that ends up broken into Pieces.
  - Result: This FE identifies the Result of an event.
  - Degree: The degree to which the fracturing is completed.

- **Cause_to_move_in_place**. An **Agent** causes a **Theme** to move with respect to a certain **Fixed_location**, generally with a certain **Periodicity**, without undergoing unbounded translational motion or significant alteration of configuration/shape.

  Frame Elements:

  - Theme: Theme identifies the object involved in motion.
  - Place: This FE identifies the Place where the event occurs
  - Direction: This frame element is used for all path-like expressions, that express the Direction of the swinging, vibrating or rotating.
  - Duration: The amount of time for which a state holds or a process is ongoing.
  - Explanation: This FE identifies the Explanation for which an event occurs.

- **Change_operational_state.** A **Device** or machine is put in (or out of) service either by a volitional **Agent** or by a **Cause** event or force acting on the Device. The **Time** when and the **Place** where the Device is put in or out of use may be specified.

  Frame Elements:

  - Device: The Device or machine that is in or out of operation.

- **Closure.** An **Agent** manipulates a **Fastener** to open or close a **Containing_object** (e.g., coat, jar). Sometimes an **Enclosed_region** or a **Container_portal** may be expressed. Since the **Manipulator** is syntactically omissible, many verbs in this frame incorporate the Fastener.
  Frame Elements:
  - Containing_Object: This FE identifies the item that is closed by the agent with the fastener.
- **Creating.** A **Cause** leads to the formation of a **Created_entity**.
  Frame Elements:
  - Created_Entity: This FE identifies the entity that the Agent intentionally creates.
  - Components: This FE identifies the Components that are attached together to form a Created_Entity.
- **Cutting.** An **Agent** cuts an **Item** into **Pieces** using an **Instrument** (which may or may not be expressed).
  Frame Elements:
  - Item: The item which is being cut into Pieces.
  - Pieces: The Pieces are the parts of the original Item which are the result of the slicing.
  - Instrument: The Instrument with which the Item is being cut into Pieces.
  - Place: Where the slicing takes place.
  - Manner: Manner in which the Item is being cut into Pieces.
  - Result: The Result of the Item being sliced into Pieces.
- **Dunking.** An **Agent** temporarily places a **Theme** into a **Substance**, often with the intention to remove it later. The **Substance** may be metonymically represented by its container. The **Theme** may be partially or completely submerged.
  Frame Elements:
  - Substance: The Substance is the fluid into which the Theme is placed. This may be metonymically represented by the container of the fluid.

  - Theme: The Theme temporarily goes partially or completely into the Substance.
  - Manner: The way that the Agent puts Theme in the Substance.

- **Filling.** These are words relating to filling containers and covering areas with some thing, things, or substance, the **Theme**. The area or container can appear as the direct object with all these verbs, and is designated **Goal** because it is the goal of motion of the Theme.

  Frame Elements:

  - Goal: The Goal is the area or container being filled. Goal is generally the NP Object in this frame.
  - Theme: The Theme is the physical object or substance which changes location.
  - Path: The Path is the trajectory of the motion
  - Place: Where the filling takes place.
  - Explanation: The Explanation for which the Agent fills or covers the Goal.

- **Grinding.** In this frame, a **Grinder** or **Grinding_cause** causes a **Patient** to be broken into smaller pieces. A **Result** or **Goal** can be present.

  Frame Elements:

  - Instrument: This FE identifies the Instrument with which the Grinder grinds the Patient.
  - Patient: The Patient is the entity which undergoes the change brought about by the Grinder or Grinding_cause.

- **Inspecting.** An **Inspector** directs his/her perceptual attention to a **Ground** to ascertain whether the Ground is intact or whether an **Unwanted_entity** is present. Alternatively, the desired outcome of the inspection may be presented as a **Purpose**.

  Frame Elements:

  - Ground: The entity that the Inspector inspects with his senses to ascertain whether it is intact.

  - Instrument: An entity used by the Inspector to scrutinize the Ground.
  - Purpose: Some of the words in this frame occur with a constituent that expresses the desired outcome of the inspection. Normally this constituent is a PP Complement headed by for, or a to-marked VP Complement.
  - Desired_state_of_affairs: The desirable state of affairs which the Inspector hopes to find to obtain when inspecting the Ground.

- **Mass_motion.** A **Mass_theme**, generally made up of many individuals, moves from a **Source** to a **Goal** with some **Path**.

  Frame Elements:

  - Mass_Theme: The mass of entities which moves.
  - Goal: Goal is the location the Mass_theme ends up in
  - Source: The Source is the location the Theme occupies initially before its change of location.
  - Manner: Any description of the motion event which is not covered by more specific FEs, including secondary effects (quietly, loudly), and general descriptions comparing events (the same way). In addition, it may indicate salient characteristics of an Mass_theme that also affect the action (presumptuously, coldly, deliberately, eagerly, carefully)

- **Placing.** An **Agent** places a **Theme** at a location, the **Goal**, which is profiled. In this frame, the Theme is under the control of the Agent at the time of its arrival at the Goal.

  Frame Elements:

  - Theme: The Theme is the object that changes location during the Placing.
  - Goal: The FE Goal is the location where the Theme ends up. This FE is profiled by words in this frame.
  - Source: The Source is the initial location of the Theme, before it changes location.
  - Area: The Area is the setting into which the Theme is placed
  - Means: The Means is an act whereby the Agent achieves the placing of the Theme.

  - Duration: The Duration is the amount of time for which the Theme is to stay in the Goal location.
  - Result: This FE identifies the Result of the Placing.
  - Explanation: The Explanation indicates why the Placing occurs.

- **Removing.** An **Agent** causes a **Theme** to move away from a location, the **Source**. The Source is profiled by the words in this frame, just as the Goal is profiled in the **Placing** frame.
  Frame Elements:
  - Theme: Theme is the object that changes location.
  - Source: The Source is the initial location of the Theme, before it changes location.
  - Explanation: The Explanation denotes a proposition from which the main clause (headed by the target) logically follows. This often means that the Explanation causes the state of affairs expressed by the target, but not in all cases.

- **Reshaping.** In this frame, a **Deformer** deforms a **Patient** possibly against a **Resistant_surface** such that it undergoes a shape-change from its canonical or original shape into the **Configuration**, a new shape.
  Frame Elements:
  - Instrument: This FE identifies the Instrument with which the Deformer affects the Patient.
  - Patient: The Patient is the entity acted on and that undergoes a change.
  - Place: The Place indicates where the Deformer intentionally affects the Patient.
  - Result: The Result is the shape the Patient becomes.
  - Manner: The FE Manner identifies the manner in which an Deformer intentionally affects the Patient.
  - Subregion: The part of the Patient which is directly affected by the re-shaping.

- **Separating.** These words refer to separating a **Whole** into **Parts**, or separating one part from another. The separation is made by an **Agent** or **Cause** and may be made on the basis of some **Criterion**.

  Frame Elements:

  - Whole: The Whole is a single entity or an aggregate of entities which is separated into Parts.
  - Parts: This refers collectively to the Parts resulting from separation of a Whole
  - Instrument: The Instrument used to cause the separation.
  - Place: Where the separation occurs.

- **Soaking.** An **Agent** places the **Theme** in the **Medium** for an extended period of time, with the intent that the Theme will be affected by and absorb some of the Medium.

  Frame Elements:

  - Theme: The Theme is the entity that changes location.
  - Medium: Medium is the substance in which the Theme is submerged.
  - Container: This FE identifies the Container that holds the Theme which is submerged and the Medium that it is in
  - Duration: The amount of time for which the Theme is soaked.
  - Place: The Place where the submerging occurs.
  - Duration: The amount of time for which the Theme is soaked.

- **State_continue.** Despite some implication that a **State** would be interrupted, the **Entity** remains in the specified **State**.

  Frame Elements:

  - Entity: A concrete or abstract Entity.
  - Place: The location where the Entity is in the State.

- **Storing.** An **Agent** has placed a **Theme** in an accessible but somewhat out-of-the-way **Location** for the purposes of maintaining it free from harm and illegitimate use while it is not being used.

  Frame Elements:

  - Theme: The Theme is the object that is kept in a Location by the Agent
  - Location: The location where the Theme is kept. Some, though not all, LUs in this frame incorporate this FE.

- **Taking.** An **Agent** removes a **Theme** from a **Source** so that it is in the Agent's possession.

  Frame Elements:

  - Theme: The Agent takes possession of the Theme.
  - Source: The location of the Theme prior to the taking.
  - Explanation: The Explanation for the taking.

- **Waiting.** A **Protagonist** delays a planned action because they cannot or do not want to proceed until an **Expected_event** occurs.

  Frame Elements:

  - Salient_Entity: A concrete or abstract entity that the Protagonist expects to participate in an Expected_event, typically that of arriving at the Place of the Protagonist.
  - End_Point: The point in time, defined either as a clock time or in terms of an event that takes place, at which the Protagonist ends the waiting.

APPENDIX

B

COOKDIAL DIALOGUES ANNOTATION

Dialogue 1 - Broccoli Cheese Crepes

- **Cause_to_be_included**. New_Member: eggs, water, all-purpose flour
- **Cause_to_be_included**. New_Member: salt
- **Cause_to_amalgamate**. Result: until smooth. **Cause_to_continue** Duration: 15 minutes
- **Taking**. Theme: saucepan. **Apply_heat**. Food: chopped onion, CoParticipant: butter, Duration: until tender
- **Cause_to_amalgamate**. Parts: all-purpose flour, Duration: until tender
- **Cause_to_amalgamate**. Parts: milk
- **Cause_change_of_phase**. Patient: mixture, Temperature_Goal: medium heat. **Cause_to_amalgamate**. Manner: constantly
- **Cause_to_amalgamate**. Duration: 2 minutes, until slightly thickened
- **Cause_temperature_change**. Item: heat, Temperature_Goal: low
- **Cause_to_be_included**. New_Member: shredded Cheddar cheese, Dijon mustard, Worcestershire sauce.
- **Cause_to_be_included**. New_Member: pepper, salt
- **Cause_to_amalgamate**. Parts: chopped broccoli. **Filling**. Goal: mixture. **Cause_to_continue**. State: warm

- **Cause_temperature_change**. Item: nonstick skillet
- **Filling**. Goal: nonstick skillet. **Cause_temperature_change**. Item: it
- **Mass_motion**. Mass_Theme: batter, Goal: into the center of skillet.
- **Cause_Motion**. Theme: pan, Explanation: to evenly coat bottom
- **Apply_heat**. Food: it, Duration: until top appears dry, 15-20 seconds longer
- **Placing**. Theme: crepes, Goal: wire rack
- **Filling**. Goal: skillet
- **Placing**. Theme: filling, Area: the center of each crepe, Goal: each crepe. **Reshaping**. Patient: each crepe
- **Placing**. Theme: seam, Area: side down, Goal: baking dish
- **Filling**. Goal: crepes, Theme: cheese
- **Apply_heat**. Food: crepes, Temperature_setting: 350 degrees F, Duration 5-7 minutes, Result: until cheese is melted.

Dialogue 2 - Cake Doughnuts

- **Taking**. Theme: bowl
- **Cause_to_be_included**. New_Member: all-purpuse flour, Existing_Member: bowl
- **Cause_to_be_included**. New_Member: white sugar, baking powder
- **Cause_to_be_included**. New_Member: salt, cinnamon, Existing_Member: batter
- **Cause_to_be_included**. New_Member: ground nutmeg
- **Cause_to_amalgamate**. Parts: these. **Reshaping**. Cause: well, Place: in the middle of it
- **Mass_motion**. Mass_Theme: milk
- **Cause_to_amalgamate**. Parts: egg. **Cause_to_be_included**. New_Member: it, Existing_Member: well

- **Cause_to_amalgamate**. Parts: egg. Means: egg beater or mixer
- **Cause_to_be_included**. New_Member: it, butter, Existing_Member: well
- **Cause_to_be_included**. New_Member: vanilla extract. **Cause_to_amalgamate**. Parts: it, Result: well blended
- **Filling**. Goal: it. **Placing**. Theme: it, Goal: refrigerator, Duration: 1 hour
- **Cause_temperature_change**. Item: fryer, Temperature_Goal: 185 degrees Celsius
- **Placing**. Theme: flour, Goal: cutting board
- **Removing**. Theme: dough butter, Source: fridge. **Placing**. Theme: it, Goal: cutting board
- **Placing**. Theme: dough, Goal: cutting board.
- **Taking**. Theme: round cutter, Explanation: so we can cut out the doughnuts. **Cutting**. Item: doughnuts
- **Cutting**. Item: holes, Place: from the center
- **Apply_Heat**. Food: doughnuts, Container: deep-fryer, Duration: until the become golden brown. **Cause_to_move_in_place**. Theme: them, Duration: one time
- **Cause_to_be_dry**: Dryee: them, Place: plates, Duration: for a little while
- **Cause_to_amalgamate**. Parts: cinnamon, sugar, Whole: bag
- **Placing**. Theme: donuts, Goal: bag, Duration: at time. **Closure**. Containing_Object: it. **Cause_to_move_in_place**. Theme: it.

Dialogue 3 - Basil Burgers

- **Cause_change_of_phase**. Item: outdoor grill, Temperature_Goal: high heat temperature
- **Taking**. Theme: bowl
- **Cause_to_be_included**. New_Member: ground beef, Existing_Member: bowl

- **Cause_to_be_included**. New_Member: Worcestershire sauce, dried basil
- **Cause_to_be_included**. New_Member: garlic salt, ground black pepper
- **Reshaping**. Patient: burger patties, Subregion: from the mixture
- **Filling**. Goal: grill. **Apply_heat**. Food: burgers, Duration: to desired doneness
- **Apply_heat**. Food: them, Duration: 6 minutes, Temperature_setting: 70 degrees Celsius
- **Placing**. Theme: cooked burgers, Goal: hamburger buns

Dialogue 4 - Dads BBQ Roast

- **Taking**. Theme: bowl
- **Cause_to_be_included**. New_Member: yellow mustard, dry onion soup mix Existing_Member: bowl
- **Taking**. Theme: aluminium foil. **Placing**. Theme: sheets Manner: crosswise
- **Taking**. Theme: sheets
- **Taking**. Theme: beef rump roast. **Absorb_Heat**. Entity: roast, Manner: dry. **Placing**. Theme: it, Area: in the center of the aluminium foil, Goal: aluminium foil.
- **Absorb_Heat**. Entity: beef, Manner: dry. **Placing**. Theme: it, Goal: aluminium foil.
- **Cause_to_amalgamate**. Parts: mustard onion soup, Place: bowl. **Filling**. Theme: mixture, Goal: roast
- **Filling**. Theme: foil, Manner: tighly, Place: around the roast, Goal: roast, Explanation: so it's all covered.
- **Taking**. Theme: aluminium foil. **Filling**. Theme: it, Place: around the roast, Goal: roast
- **Placing** Theme: outdoor grill

- **Apply_heat**. Theme: charcoal. **Cause_to_move_in_place**. Theme: it, Place: to one side of the grill, Goal: grill. **Placing** Theme: roast, Goal: grill. **Filling**. Theme: it
- **State_continue**. Entity: roast, coals, State: all over the grill, on one side Place: grill
- **Apply_heat**. Food: roast, Duration: 2 hours. **Cause_to_move_in_place**. Periodicity: every 30 to 45 minutes
- **Cause_temperature_change**. Item: roast, Duration: 10 minutes
- **Placing** Theme: roast, Goal: serving plate
- **Filling**. Theme: juices, mustard rub, Goal: beef slices

Dialogue 5 - Broiled Slow Roasted Butterflied Leg of Lamb With Cumin and Garlic

- **Cause_to_amalgamate**. Parts: oil, garlic, salt, pepper, cumin, oregano,
- **Filling**. Theme: mixture, Goal: lamb, Area: on both side of the lamb. **Waiting**. Salient_Entity: lamb, Duration: one hour, End_Point: room temperature
- **Waiting**. Salient_Entity: lamb, Duration: one hour, End_Point: room temperature
- **Cause_temperature_change**. Item: oven, Temperature_Goal: high heat, Duration: 10 minutes
- **Filling**. Theme: foil, Goal: roasting pan
- **Placing**. Theme: lamb, it Goal: roasting pan, wire rack, Place: oven
- **Apply_heat**. Food: lamb, Duration: 8 minutes. **Cause_to_move_in_place**. Theme: pan, Manner: regularly, Explanation: so the entire surface browns evenly
- **Cause_to_move_in_place**. Theme: lamb, pan; Manner: regularly, Explanation: so the other surface is facing upwards.

- **Change_operational_state**. Device: broiler. **State_continue**. Entity: lamb, Place: outside the oven, Duration: 10 minutes
- **Cause_temperature_change**. Item: oven; Temperature_Goal: 325 degree Fahrenheit
- **Placing**. Theme: thermometer, Area: into the thickest portion of the lamb; Goal: lamb
- **Apply_heat**. Theme: lamb; Duration: until the thermometer shows 140 degrees, 50 minutes to 1 hour
- **Removing**. Theme: lamb, Source: oven. **Filling**. Theme: lemon juice, fresh herbs
- **Cutting**. Item: lamb. **Placing**. Goal: platter. **Filling**. Goal: lamb, Theme: juices; Source: lamb

Dialogue 6 - Cinnamon Bread I

- **Cause_temperature_change**. Item: oven; Temperature_Goal: 175 degree Celsius
- **Filling**. Theme: butter, Goal: loaf pan
- **Cause_to_be_included**. New_Member: white sugar
- **Cause_to_be_included**. New_Member: baking powder
- **Cause_to_be_included**. New_Member: baking soda, cinnamon; Existing_Member: mix
- **Cause_to_be_included**. New_Member: salt, buttermilk, vegetable oil; Existing_Member: mix
- **Cause_to_be_included**. New_Member: eggs, vanilla extract; Existing_Member: mix
- **Cause_to_amalgamate**. Whole: it; Duration: 3 minutes
- **Placing**. Theme: it; Goal: loaf pan. **Reshaping**. Subregion: top

- **Taking**. Theme: bowl. **Cause_to_amalgamate**. New_Member: white sugar, cinnamon, butter
- **Cause_to_amalgamate**. Whole: ingredients; Place: small bowl; Result: until they are crumbly
- **Filling**. Theme: crumbled topping; Goal: smoothed batter
- **Taking**. Theme: knife. **Cutting**. Manner: in a light swirling motion; Place: in the batter; Item: batter
- **Apply_heat**. Food: cake; Duration: 50 minutes; Cooking_Appliance: oven
- **Apply_heat**. Food: it, Duration: 35 minutes more
- **Inspecting**. Ground: it; Purpose: if it's ready
- **Removing**. Theme: bread. **Cause_temperature_change**. Item: it.

Dialogue 7 - Baked Chicken Breasts Supreme

- **Taking**. Theme: bowl; Purpose: to create a mixture
- **Cause_to_be_included**. New_Member: sour cream
- **Cause_to_be_included**. New_Member: lemon juice, Worcestershire sauce
- **Cause_to_be_included**. New_Member: celery seed, Hungarian sweet paprika
- **Cause_to_be_included**. New_Member: minced garlic glove
- **Cause_to_be_included**. New_Member: salt, pepper
- **Placing**. Theme: chicken; Goal: mixture. **Cause_to_move_in_place**. Theme: chicken; Explanation: so everything is coated with the mixture
- **Filling**. Theme: it. **Soaking**. Theme: it; Place: refrigerator; Duration: during the night
- **Removing**. Theme: chicken; Source: marinade. **Filling**. Goal: chicken; Theme: crumbs
- **Filling**. Theme: crumbs; Goal: chicken

- **Placing**. Theme: chicken; Goal: baking pan
- **Apply_Heat**. Food: it; Temperature_Goal: 350 degrees Fahrenheit; Duration: 45 minutes

Dialogue 8 - Chocolate Hazelnut Fruit Crepes

- **Taking**. Theme: readymade crepes, chocolate hazelnut spread
- **Filling**. Theme: hazelnut spread; Goal: crepe
- **Taking**. Theme: bananas. **Cutting**. Theme: them
- **Placing**. Theme: sliced banana; Area: on the center of each crepe; Goal: crepe
- **Reshaping**. Patient: crepes. **Placing**. Theme: them; Place: skillet; Duration: 1,5 minutes
- **Placing**. Theme: it; Goal: plates; Means: with some whipped cream

Dialogue 9 - Dziriat Algerian Almond Tarts

- **Cause_change_of_phase**. Patient: water
- **Removing**. Theme: water; Source: heat. **Cause_to_be_included**. New_Member: raw almonds
- **Cause_to_be_included**. Existing_Member: them; Place: water. **Soaking**. Theme: them; Duration: 5 minutes
- **Emptying**. Source: almonds. **Emptying**. Source: them
- **Filling**. Theme: almonds; Goal: baking sheets. **Apply_Heat**. Food: them; Duration: 95 degrees Celsius
- **Apply_Heat**. Duration: until they are completely dry and toasted
- **Placing**. Theme: them; Place: aside
- **Taking**. Theme: saucepan. **Cause_to_amalgamate**. Parts: sugar, water. **Cause_change_of_phase**. Patient: it

- **Cause_temperature_change**. Temperature_setting: low heat. **Cause_to_be_included**. New_Member: lemon juice. **Absorb_Heat**. Entity: it; Duration: 30 to 40 minutes; Results: until it becomes syrupy
- **Cause_to_be_included**. New_Member: lemon juice
- **Cause_to_amalgamate**. Parts: orange blossom water. **Removing**. Theme: it; Source: heat
- **Placing**. Theme: syrup; Place: aside
- **Taking**. Theme: bowl. **Cause_to_amalgamate**. Parts: all-purpose flour, salt
- **Reshaping**. Cause: hole; Place: in the center of the mixture; Patient: mixture. **Cause_to_be_included**. Place: hole
- **Cause_to_be_included**. New_Member: vegetable oil, egg
- **Cause_to_be_included**. New_Member: lemon juice, orange blossom water
- **Cause_to_amalgamate**. Whole: everything; Instrument: fingers; Duration: until it looks like coarse crumbs
- **Filling**. Goal: dough; Theme: water; Manner: gradually. **Cause_to_amalgamate**. Whole: it; Duration: until becomes soft and pliable
- **Separating**. Whole: dough; Parts: portions. **Filling**. Goal: it, Theme: wet cloth
- **Placing**. Theme: it; Place: aside. **Taking**. Theme: almonds
- **Placing**. Theme: them; Goal: food processor. **Grinding**. Patient: them; Manner: finely
- **Taking**. Theme: grinded almonds. **Placing**. Place: mixing bowl
- **Cause_to_be_included**. New_Member: sugar, baking powder
- **Cause_to_be_included**. New_Member: vanilla powder, zested lemon
- **Cause_to_be_included**. New_Member: orange flower water
- **Cause_to_amalgamate**. Parts: eggs. **Cause_to_be_included**. New_Member: eggs; Manner: one at time. **Cause_to_amalgamate**. Manner: constantly; Result: until you get paste-like mixture

- **Taking**. Theme: dough
- **Reshaping**. Patient: dough. **Filling**. Theme: cornstarch; Subregion: rolling surface; Explanation: in order to prevent sticking
- **Filling**. Theme: cornstarch
- **Cutting**. Pieces: circles; Place: dough
- **Filling**. Theme: cornstarch; Goal: surface of each circle. **Placing**. Goal: tart mold
- **Reshaping**. Patient: circles; Subregion: onto the sides and the bottom; Place: tart mold; Explanation: so it is fitting perfectly. **Cutting**. Item: tart mold; Pieces: edges; Place: at the rim
- **Removing**. Patient: extra dough; Place: over the rim; Source: tart mold
- **Filling**. Theme: almond filling; Goal: mold
- **Apply_Heat**. Food: them; Place: on the top shelf; Cooking_Appliance: oven; Temperature_Setting: 175 degree Celsius; Duration 20 to 25 minutes
- **Removing**. Theme: them, tarts ; Source: oven, molds
- **Dunking**. Theme: them; Manner: directly; Substance: sugar syrup
- **Placing**. Theme: pine nut; Goal: tart; Explanation: for decoration
- **Placing**. Theme: tarts; Goal: wire rack; Explanation: so they can drain

Dialogue 10 - Basic Syrup for Sunset Cooler

- **Taking**. Theme: saucepan
- **Cause_to_be_included**. New_Member: white sugar, water; Existing_Member: pan
- **Cause_to_be_included**. New_Member: pandan leaves
- **Cause_change_of_phase**. Patient: it
- **Cause_temperature_change**. Item: heat; Result: until the sugar is dissolved

- **Removing**. Theme: pandan leaves
- **Removing**. Theme: syrup. **Cause_temperature_change**. Item: it
- **Closure**. Containing_Object: it; Place: bottle. **Placing**. Source: refrigerator

APPENDIX

C

PYTHON SCRIPTS

```python
import json
import os

# Define the path where the CookDial data is stored
path = 'Data/CookDial'

# Walk through all files in the directory and its subdirectories
for subdir, dirs, files in os.walk(path):
    # Iterate over each file that ends with ".json"
    for file in [x for x in files if x.endswith(".json")]:
        outJson = {}  # Initialize an empty dictionary to store the output data

        # Open and load the JSON file
        messages = json.load(open(path + '/' + file))['messages']
        outMessages = []  # Initialize an empty list to store the processed messages

        # Loop through each message in the 'messages' key
        for message in messages:
            # Check if the message is from the Bot
            if message['bot']:
                # Append the message text and author as 'Bot' to the outMessages list
                outMessages.append({'text': message['utterance'], 'author': 'Bot'})
            else:
                # Otherwise, append the message text and author as 'Human'
                outMessages.append({'text': message['utterance'], 'author': 'Human'})

        # Add the processed messages and file ID to the output JSON
```

```
        outJson['data'] = {'id': file, 'dialogue': outMessages}

        # Write the processed data into a new JSON file in the 'Processed' folder
        with open(path + '/Processed/' + file, "w") as outfile:
            # Convert the Python dictionary to a JSON string and write it to the file
            outfile.write(json.dumps(outJson, indent=4))  # Pretty print the JSON with indentation
```

LISTING C.1: Python code for CookDial Annotation Task

```
import json
import pandas as pd
from nltk import agreement
from nltk.metrics.distance import masi_distance
from nltk.metrics.distance import jaccard_distance

# Open file
with open("annotation_1305.json", "r") as read_file:
    data = json.load(read_file)

# List of Frame Intents
intents = ['apply_heat', 'Cause_to_amalgamate', 'Cause_to_be_included', 'Dunking',
           'Placing', 'Reshaping', 'Separating', 'Removing', 'Cutting', 'Grinding',
           'Taking', 'Cause_change_of_phase', 'Mass_motion', 'Filling', 'Cause_motion',
           'Absorb_heat', 'Cause_to_continue', 'Cause_to_move_in_place', 'Closure',
           'Cause_to_be_dry', 'Cause_temperature_change', 'Soaking', 'State_continue',
           'Change_operational_state', 'Inspecting', 'Emptying', 'Storing', 'Waiting',
           'Cause_to_fragment']

data_sentences = []

# Construct dataframe from json
for datum in data:
    id_dialogo = datum['data']['id']
    # For each label in the annotations
    for label in datum['annotations']:
        for utterance in label['result']:
            # If the annotation is an intent
            if utterance['value']['paragraphlabels'][0] in intents:
                # Add information about the intent to the final list
                data_sentences.append({'id_dialogo': id_dialogo,
```

```
                                        'annotator_id': label['completed_by'
                                            ],
                                        'frame': utterance['value']['
                                            paragraphlabels'][0],
                                        'turn': utterance['value']['start'],
                                        'text': utterance['value']['text'].
                                            strip()})

# Convert the list of dictionaries to a DataFrame
df = pd.DataFrame(data_sentences)

# Only keep dialogues where annotator 15 is present
df = df[df['id_dialogo'].isin(list(df[df['annotator_id'] == 15]['id_dialogo'
    ]))]

# Create a unique label identifying dialogue and turn
df['annotation_id'] = df.apply(lambda x: x['id_dialogo'][:-5] + "_" + x['
    turn'], axis=1)

# Remove unnecessary columns
df = df.drop(['id_dialogo', 'turn', 'text'], axis=1)

# Create a list of annotations for each turn
df = df.groupby(['annotation_id', 'annotator_id'])['frame'].apply(list)

# Extract the indexes representing the annotation id
ids = df.index.values.tolist()

# Create a list of annotations
annots = []
for i, row in enumerate(df):
    annots.append([ids[i][1], ids[i][0], frozenset(row)])

# Create the NLTK tasks to compute distance between annotations
jaccard_task = agreement.AnnotationTask(distance=jaccard_distance)
masi_task = agreement.AnnotationTask(distance=masi_distance)
tasks = [jaccard_task, masi_task]

# Compute Krippendorf agreement
for task in tasks:
    task.load_array(annots)
    print("Statistics for dataset using {}".format(task.distance))
    print("Alpha: {}".format(task.alpha()))
```

LISTING C.2: Python code for Annotation Agreement

```
# FI and FE dictionaries
intents = ['apply_heat', 'Cause_to_amalgamate', 'Cause_to_be_included', '
    Dunking', 'Placing', 'Reshaping', 'Separating', 'Removing', 'Cutting', '
    Grinding', 'Taking', 'Cause_change_of_phase', 'Mass_motion', 'Filling',
    'Cause_motion', 'Absorb_heat', 'Cause_to_continue', '
    Cause_to_move_in_place', 'Closure', 'Cause_to_be_dry', '
    Cause_temperature_change', 'Soaking', 'State_continue', '
    Change_operational_state', 'Inspecting', 'Emptying', 'Storing', 'Waiting
    ', 'Cause_to_fragment' ]

elements = ['TemperatureSetting', 'Heating_instrument', 'Container', '
    Duration', 'Medium', 'Place', 'Parts', 'Whole', 'Means', 'Time', '
    ExistingMember', 'NewMember', 'Group', 'Item', 'Pieces', 'Instrument', '
    Manner', 'Result', 'Substance', 'Theme', 'Patient', 'Goal', 'Source', '
    Area', 'Explanation', 'Subregion', 'Degree', 'MassTheme', 'Path', '
    Entity', 'HeatSource', 'Medium', 'State', 'Direction', 'ContainingObject
    ', 'Dryee', 'TemperatureGoal', 'Device', 'Ground', 'Purpose', 'Alterant'
    , 'Desired_state_of_affairs', 'ProducedFood', 'Cotheme', 'CoParticipant'
    , 'Periodicity', 'SalientEntity', 'EndPoint', 'Cause']

# set the variables to count FI & FE. This script serves to identify the IFs
     and FEs for each turn
elementsCountersMap = {}
intentsCountersMap = {}
# enter the data folder
intentsNormalisedPositionsFinal = []  # list of dialogues
# identify the result folder
for datum in data:
   intentsNormalisedPositions = []  # empty list where the calculation of
       each dialogue is added
   maxTurn = 0  # max turn of dialogues
   for label in datum['annotations'][0]['result']:
        # in the 'paragraphlabels' folder in the 'intents' dictionary of
            which we take the first string (0)
        if label['value']['paragraphlabels'][0] in intents:
            # prints the corresponding value of the string '0' by naming it
                'name_intent'.
            print('name_intent: ' + label['value']['paragraphlabels'][0])
            print('turn: ' + str(label['value']['start']))
            print('StartOffset:' + str(label['value']['startOffset']))
            print('EndOffset: ' + str(label['value']['endOffset']))
            print('Text: ' + str(label['value']['text']))

            intentsNormalisedPositions.append((label['value']['
                paragraphlabels'][0], int(label['value']['start'])))  # add
                the recognised intent and the shift in which it appears to
                the list of the individual dialogue
```

```
26              maxTurn = int(label['value']['start']) if int(label['value']['
                    start']) > maxTurn else maxTurn  # if the shift is greater
                    than the one found so far, it becomes the maximum shift,
                    otherwise the maximum remains the same
27
28              # This serves for calculating the distribution of FE. for each
                    data folder, it finds the annotations folder and takes the
                    first element (0), where the dictionary 'result' is located
29               for inLabel in datum['annotations'][0]['result']:
30                  if inLabel['value']['paragraphlabels'][0] in elements and
                        label['value']['start'] == inLabel['value']['start'] and
                         inLabel['value']['startOffset'] >= label['value']['
                        startOffset'] and inLabel['value']['endOffset'] <= label
                        ['value']['endOffset']:
31                   print as "name_elements" values contained in paragraphlabels
32                   print('    name_elements: ' + inLabel['value']['paragraphlabels
                         '][0])
33                   print('    turn: ' + str(inLabel['value']['start']))
34                   print('    StartOffset:' + str(inLabel['value']['startOffset'])
                         )
35                   print('    EndOffset: ' + str(inLabel['value']['endOffset']))
36                   print('    Text: ' + str(inLabel['value']['text']))
37
38  # divide each n of shifts by the maximum n of shifts
39      intentsNormalisedPositions = [(x[0], x[1]/maxTurn) for x in
            intentsNormalisedPositions]
40  # add the list of the individual dialogue to the list of all dialogues
41      intentsNormalisedPositionsFinal.append(intentsNormalisedPositions)
42
43  # prepares the list for transforming into a dataframe
44  intentsNormalisedPositionsFinal = [(x[0], x[1]) for y in
        intentsNormalisedPositionsFinal for x in y]
45  # dataframe
46  df = pd.DataFrame(intentsNormalisedPositionsFinal)
47  # save dataframe in tsv
48  df.to_csv('FIntents_Distribution.csv', decimal=",", sep="\t")
```

Listing C.3: Python code for Frame Intents Distribution


UNIONE EUROPEA
Fondo Sociale Europeo


Ministero dell'Università
e della Ricerca


PON
RICERCA
E INNOVAZIONE


REACT EU

Acknowledgements

It has been a great privilege to embark on this academic journey, which has been fundamental in shaping both my professional achievements and my personal growth.

I thank my supervisors, Prof. Marco Venuti and Prof. Francesco Cutugno, who placed their trust in me from the very beginning, supporting both my ideas and my professional development.

I thank the entire URBAN/ECO research group for welcoming me and giving me professional advice. In this respect, I thank Prof. Antonio Origlia for his help in the development of the computational part of this work, along with his staff.

I thank my colleagues at the University of Catania for the enriching collaborations during my PhD, where the multidisciplinary nature of our work significantly enhanced my knowledge across various fields.

I thank Prof. Dr.-Ing. Hendrik Buschmeier and the entire LiLiLab group for welcoming me with enthusiasm and warmth during my stay abroad at the University of Bielefeld. Ich danke Ihnen herzlich für ihre Unterstützung und fühle mich sehr geehrt, Sie kennengelernt zu haben.

I thank the entire staff at Logogramma s.r.l., particularly Prof. Dr. Valentina Russo and Prof. Dr. Azzurra Mancini, for the opportunity to carry out my industrial internship, immersing me in the corporate world — a dynamic environment distinct from academia — and allowing me to engage in various activities.

Lastly, I thank my parents, who have supported me with unwavering trust from day one. Your constant love and guidance have been fundamental to my life, and I will be eternally grateful for your role in helping me reach this point and for the strength you have given me along the way.

I am sincerely grateful for everything I have learned so far, as well as for what remains to be discovered, as

The only true wisdom is in knowing you know nothing - Socrates.