UNIVERSITÀ DI NAPOLI "FEDERICO II"

Tesi di Dottorato

Corso di Dottorato in Mind, Gender and Language

XXXIII Ciclo

- Borsa di dottorato innovativo a caratterizzazione industriale -

"Shouldn't I use a polar question?" Proper Question Forms Disentangling Inconsistencies in Dialogue Systems

Candidata

Maria Di Maro

Tutor interno – Università degli Studi di Napoli "Federico II"

Prof. Francesco Cutugno

Tutor estero – University of Bielefeld

Prof. Stefan Kopp

Tutor aziendale – Interactive Media S.p.a.

Dott. Paolo Turriziani

Coordinatore del Dottorato

Prof. Dario Bacchini

Anno Accademico 2020

SIRI

UNIONE EUROPEA
Fondo Sociale Europeo
Fondo Europeo di Sviluppo Regionale

Ministero dell'Istruzione, dell'Università e della Ricerca

PON
RICERCA E INNOVAZIONE
2014 - 2020

Furuike ya
kawazu tobikomu
mizo no oto

Matsuo Bashō

Abstract

This work reports on the description of a specific class of clarification requests, adopted for the negotiation of pieces of information part of the common ground for argumentation strategies in human-machine interaction. Two studies are carried out to prove the adequateness of a specific form of polar question in a specific pragmatic situation, where a presupposition is contradicted by a new evidence. Whereas the first one proves the appropriateness of the negative form, the second one also demonstrate how the use of such a form, in the aforementioned pragmatic situation, can affect the principle of robustness, in terms of observability and recoverability, important in human–machine interaction applications. Given the results obtained in the two studies, dialogue systems with such capabilities are, therefore, a desirable goal, as they are expected to lead to improved usability and naturalness in conversation. For this reason, I present here a system capable of detecting conflicts and of using argumentation strategies to signal them consistently with previous observations.

In dieser Arbeit wird über die Beschreibung einer spezifischen Klasse von Klärungsfragen berichtet, die für die Aushandlung von Informationen eingesetzt werden, die zur gemeinsamen Grundlage für Argumentationsstrategien in der Mensch-Maschine-Interaktion gehören. Es werden zwei Studien durchgeführt, um die Angemessenheit einer bestimmten Form von Entscheidungsfrage in einer spezifischen pragmatischen Situation zu beweisen, in der eine Voraussetzung durch einen neuen Beweis widerlegt wird. Während die erste die Angemessenheit der Negativform beweist, zeigt die zweite auch, wie sich die Verwendung einer solchen Form in der genannten pragmatischen Situation auf das Prinzip der Robustheit im Sinne von Beobachtbarkeit und Wiederherstellbarkeit auswirken kann, das in Anwendungen der Mensch-Maschine-Interaktion wichtig ist. Angesichts der in den beiden Studien erzielten Ergebnisse sind Dialogsysteme mit solchen Fähigkeiten daher ein wünschenswertes Ziel, da sie zu einer verbesserten Benutzerfreundlichkeit und Natürlichkeit in der Konversation führen sollen. Aus diesem Grund stelle ich hier ein System vor, das Konflikte erkennt und, mit Hilfe von Argumentationsstrategien, signalisiert.

Questo lavoro riporta la descrizione di una specifica classe di richieste di chiarimento, adottate per la negoziazione di informazioni parte del common ground nell'ambito delle strategie di argomentazione nell'interazione uomo-macchina. Due studi sono condotti per dimostrare l'adeguatezza di una specifica forma di domanda polare nella situazione pragmatica per cui una presupposizione è contraddetta da una nuova evidenza. Mentre il primo dimostra l'adeguatezza della forma negativa, il secondo dimostra anche come l'uso di tale forma nella suddetta situazione pragmatica possa incidere sul principio di robustezza, in termini di osservabilità e recuperabilità, importante nelle applicazioni di interazione uomo-macchina. Partendo dai risultati ottenuti nei due studi sopra menzionati, vengono qui supportate le basi per il perseguimento dell'obiettivo di sviluppare sistemi di dialogo con abilità pragmatiche che permettano di incrementare i livelli di usabilità e naturalezza nella conversazione. Per questo motivo, presento qui un sistema in grado di rilevare i conflitti e di utilizzare strategie argomentative per segnalarne presenta e natura coerentemente con le osservazioni precedenti.

Table of Contents

List of Figures

List of Tables

Publications

Some of the work presented in this thesis has previously appeared in, or is due to appear in, various conference proceedings and journals. Other works can be found on Google Scholar[1]. The list below refers to the different chapters where the papers were mentioned.

Chapter 1

Maria Di Maro, Marco Valentino, Anna Riccio, and Antonio Origlia. "Graph Databases for Designing High-Performance Speech Recognition Grammars". In: *IWCS 2017. 12th International Conference on Computational Semantics—Short papers* (2017).

Maria Di Maro, Sara Falcone, and Francesco Cutugno. "Prosodic Analysis in Human-Machine Interaction". In: *Studi AISV1* (2018), pp. 227-239.

Chapter 3

Maria Di Maro, Mohamed Diaoulé Diallo, and Francesco Cutugno. "Information-Processing Machines and the Access-Conscious Recognition of Common Ground In-consistencies: A Proposal." In: *Proceedings of the Second Symposium on Psychology-based Technologies* (2020)

Chapter 4

Maria Di Maro, Antonio Origlia, and Francesco Cutugno. "PolarExpress: Polar question forms expressing bias-evidence conflicts in Italian". In: *International Journal of Linguistics* (2021) – *submitted.*

Chapter 5

Maria Di Maro, Antonio Origlia, and Francesco Cutugno. "Solving Common Ground inconsistencies: the role of polar question forms in human-machine

[1] https://scholar.google.com/citations?user=Yslr-U4AAAAJhl=it

interaction". In: *Computer Speech and Language* (2021) – *submitted.*

Chapter 6

Maria Di Maro, Antonio Origlia, and Francesco Cutugno. "Overview of the EVALITA2018 Spoken Utterances Guiding Chef's Assistant Robots (SUGAR) Task". In: *Proceedings of EVALITA 2018* (2018).

Maria Di Maro, Antonio Origlia, and Francesco Cutugno. "Cutting melted butter? Common Ground inconsistencies management in dialogue systems using graph databases". In: *Italian Journal of Computational Linguistics, Special Issue on Computational Dialogue Modelling* (2021) – *submitted.*

Introduction

This work deals with inconsistencies management in dialogue systems. More specifically, among various typologies of dialogue systems, I will take into account the ones where interaction tends to construct a meaning and a goal in consecutive steps, in opposition to the systems where the understanding of single separated commands, all independent from the previous one, occurs. The interaction designed in these systems generates a sequence of intents (for a definition of *intent* see Chapter 6) which, incrementally, leads to the definition of a higher level goal: actors of the dialogue (typically a human being and an artificial character) share a common knowledge, or *Common Ground* (Chapter 2), a context, and the main objective of the interaction, that is letting the artificial counterpart of the dialogue perform a complex task based on the sequential inputs produced by the user. In this context, the system ability to check for consistency and coherence of the data contained in the sequence of elementary intents aiming at building the final complex goal, becomes indispensable. Therefore, when facing consistency problems, the formulation of clarification requests (CRs), with the consequent application of language generation and speech synthesis algorithms, becomes a matter of research. The objective of this work is to highlight the nature of specific clarification requests, in that not only are they used to express a conceptual and/or informative inconsistency toward a presupposition, but they can also denote the kind of problem and the mental state of the speaker toward the problem itself.

The present work aims therefore at answering three specific research questions:

RQ1 Which forms of clarification requests are frequently adopted by speakers when Common Ground Inconsistencies occur?

RQ2 Do Common Ground Inconsistencies require specific superficial polar question forms in Italian as well as in English and German?

RQ3 Does using a specific polar question form result in an improved communication efficiency, or is it just a matter of naturalness?

RQ4 How can Common Ground Inconsistencies be detected and signalised in computational architectures for dialogue management systems?

Starting from the proposal of a general fine-grained classification and description of clarification requests based on related work and corpora analysis, I focus on the description of a specific class of clarification requests, concerning the negotiation of information, part of the communal and/or personal common ground, whose modelling is important in human–machine interaction applications for the correct interpretation of the system's internal state and the consequent adoption of the appropriate conflict resolution by the user. To do that, common ground clarification requests were firstly analysed, as far as form, bias, and evidence are concerned. Since polar questions were found to be mostly used to resolve possible cognitive inconsistencies, the relationship between polar questions forms and the bias they express was taken into account, on the basis of previous work by Domaneschi et al. [56]. Therefore, the contrast between bias and evidence was analysed starting from the examples extracted from a German available corpus. This was useful to understand which form is generally preferred to express a specific contrast while conversing (i.e., positive polar questions, high negation polar questions, low negation polar questions, really-positive polar questions). After collecting the forms distribution across bias and evidence contrasts, a first narration-based experiment was carried out to study the syntax-pragmatics interface in the use of polar questions given different conflict scenarios in the Italian language. This experiment resulted in a clear tendency for preferring high negation polar questions in the past tense when a positive bias clashes with a negative contextual evidence, pointing out the importance speakers give to the syntactic form with respect to pragmatic needs. Starting from these results, a second study was planned to put participants not in a narrated conflicting situation, but in a real conflicting context. This was adopted to simulate what happens in human-machine interaction and how the previously considered forms would be considered appropriate to improve the interaction quality. Specifically, it was hypothesised that in case of conflict between the user believes (bias) and contextual observations (evidence), the use of a negative polar question increases the *observability* degree of the internal state of the dialogue system and decreases the required time for the recovery of the inconsistency problem. This was important to motivate the development of a dialogue system architecture capable of dynamically constructing and representing the Common Ground, in the form of a graph database, whose adoption was aimed at making the system capable of noticing possible conflicts in the Common Ground, and of signalling them in an efficient way. To do so, a *Conflict Search Graph* (Chapter 6) was used to let the system be aware of the conflict. Specifically, attention will be drawn upon the Conflict Search Graph as part of the architecture of a dialogue system, with insights on its structure and on its ability to store knowledge, to recognise problems, and to make them explicit via clarification requests signalling inconsistencies, i.e., Common Ground CRs in the form of polar questions.

In order to answer these research questions, the work is organised as follows:

Chapter I It introduces the dialogue systems' field of study with its state of the art and theoretical challenges, mainly concerned with pragmatics and its cognitive aspects along with its computational applications. Motivations and objectives are, therefore, highlighted.

Chapter II It deals with the theory of *grounding* and the adoption of argumentation strategies in conflicting scenarios, in human-human as well as in human-machine interaction. Moreover, it introduces an in-depth classification of CRs, as one of the possible grounding tools. This represents the first original contribution of this work. More details will be especially given on one of the classes described, namely the Common Ground Clarification Requests and their most common syntactic form, i.e., Polar Questions. Concerning Polar Questions, the concepts of *bias* and *evidence* will be described.

Chapter III This chapter describes the corpus analysis needed to deduce the type of Clarification Requests mainly adopted in deliberation dialogues. The results collected from this analysis, concerned with the use of Common Ground Clarification Requests, are the foundation of the theoretical framework for a computational model, whose hypotheses and motivations will guide the experiments of the next chapters. This chapter aims at answering to RQ1 through data analysis.

Chapter IV This chapter describes the first experiment on the use of specific forms of polar questions as an appropriate linguistic strategy in Common Ground Inconsistencies, in order to answer to RQ2. Based on an experiment carried out for English and German languages, a more complex version is presented. Modifications to the original protocol allowed to capture nuances in the subjects' evaluations of appropriateness towards the considered forms.

Chapter V In this chapter, a second experiment was built to answer RQ3. In detail, starting from the results collected from the experiment in Chapter IV, this second experiment aimed at putting speakers in natural conflicting situations, where a positive bias, i.e., the current presuppositional mental state of the participant, comes in conflict with a negative contextual evidence. This was useful to understand whether the choice of a specific form of Common

Ground CR was mainly dictated by a matter of naturalness and perceived appropriateness or whether the use of a specific form could help the efficiency of the conversational exchange, in terms of interaction principles, and specifically *observability* and *recoverability*.

Chapter VI The final chapter presents the application of the results collected from the previous experiments in the design of an argumentation-based software architecture for dialogue management, mainly concentrating on the ability to recognise information inconsistencies and to signal them coherently with previous observations. This is intended to answer RQ4. Details on general system performances are also given.

Chapter 1

Introductory Theories on Pragmatic Models of Conflicting Representations in Conversation

This chapter introduces the theoretical frame of interest of this work. Starting with an introduction to dialogue systems, i.e., what they are, what they are made up of, how they work, how to develop them, challenges and application leaks are described. In fact, less attention has been given in the last years to some aspects of pragmatics, a branch of linguistics whose aim is to study language in its context of use, namely in conversation. One of these aspects is the nature and identification of conflicts between presuppositions and evidences. Such conflicts representations in conversation, along with their modelling in dialogue systems, can be framed in the field of computational pragmatics, that studies the encoding and decoding of mental processes in the interlocutors while conversing. This framework is reported here and provides support to the motivation of this research.

1.1 Dialogue Systems: Modules, Training, and Challenges

This section introduces the application system of the theory and modelling proposed in this thesis, namely dialogue systems. Insights will be given on their structure, modules, techniques, and on-going challenges.

Dialogue systems, also referred to as conversational agents, are nowadays in the spotlight in different commercial, academic and industrial sectors: it will suffice to consider the success and popularity of tools like Amazon Alexa and Google Home [99], or of the widespread in-car dialogue systems [15, 87]. Conversational agents are computer systems capable of interacting with humans through verbal signals. They are one of the most currently researched field in Artificial Intelligence, since the ability to communicate one's understanding by means of language is one possible way to manifest intelligence. While a shared opinion of how *intelligence* can be defined is far from being widely accepted, one possible definition is proposed in the Macmillan Dictionary[1], which defines it as "the ability to understand and think about things, and to gain and use knowledge". In this definition, one concept draws particular attention: 'knowledge'. Building the knowledge base for such systems is indeed the first step to give them intelligence. For this particular goal, the use of some tools facilitates the job of interaction designers, such as linguists. Concerning the learning approaches used in such systems, at the two extremes of the learning continuum, we find, on the one hand, deterministic rules given to the system to interpret some particular signals and react to them appropriately [106], whereas on the other hand we have end-to-end dialogue systems which do not make any distinction in the abilities the system should perform at different levels, but it is provided with data from which tendencies are statistically extracted [126, 167, 145, 21]. In the middle, we have the possibility to train different modules with the application of different strategies and tools. Overall, in the field of language understanding and generation, the corpus-driven approach is becoming increasingly important to infer, with the application of different machine learning algorithms, knowledge and communicative strategies [144]. This means that appropriate collections of data, in combination with specific tools, are required to model one's own system.

As anticipated, dialogue systems are interactive devices. *Interacting* means "to act, or have some effect on each other"[2]. The mutual influence agents can have on one another is built through communicative processes, both verbal and not verbal. On the other hand, *communicating* means to transmit information. According to the Shannon–Weaver model of communication, mostly applicable to machines' interaction, communication deals with the transmission of signals from one system to another, where the system communicating can be of the same nature or not [146]. According to this model, the transmitter encodes a message which is sent through a channel to the receiver who decodes it. The communication channel is also called *noise* because it can be loaded with noise of different kind. Nevertheless, communication is more than just transmitting information, as information must be processed in order to enable the receiving agent to produce a coherent output (see Chapter 3 for details about information-processing machines). Moreover, as stated by Allwood [2], com-

[1] Macmillan Dictionary Online: https://www.macmillandictionary.com/ [last consultation on 12th December 2020]

[2] Cambridge Dictionary Online: https://dictionary.cambridge.org/dictionary/ [last consultation on 12th December 2020]

munication includes not only the sharing of information, but also of cognitive content or understanding with varying degrees of awareness and intentionality. In fact, A and B communicate if and only if A and B share a cognitive content as a result of A's influencing B's perception, understanding and interpretation and B's influencing A's perception, understanding and interpretation. Despite its little applicability in human conversation, Shannon and Weaver's model is useful to understand how communication works in terms of processes' states. This model can indeed be compared with the one described by Jakobson about the functions of language [79]. According to the author, in fact, the elements interacting in communication are i) the addresser, who sends a message to the addressee; ii) the message, which is connected and interpretable because of the presence of a context it can refer to; iii) a code, common to the addresser and addressee, used to codify the message; iv) a contact, which is the physical channel and the psychological connection between the addresser and the addressee, enabling both of them to enter and stay in communication. To each item of the communication circuit corresponds a specific language function:

context The referential function corresponds to the contextual referent described in the message or which the message refers to, such as a situation, an object, or a mental state. This function can be applied through both definite descriptions and deictic words.

message The poetic function focuses on the message and is the operative function in poetry as well as slogans.

addresser The emotive function relates to the addresser and is unfolded by interjections and other vocalisations that add information concerning the speaker's internal state.

addressee The conative function directly engages the addressee by using, for instance, vocatives and imperatives.

contact The phatic function refers to the use of language for the sake of the interaction and is therefore linked to the contact, also called channel. One common application of this function is exemplified by communicative strategies used to open, maintain, verify or close the communication channel. In fact, being this channel subject to noise, as also mentioned by Shannon and Weaver, such as lack of attention, environmental noise, ambiguities, etc., it is important to check that the information crossed the channel without problem and that the addressee understood it correctly. This language function is studied in the field of pragmatics when processes concerned with grounding occur.

code The metalingual or metalinguistic function is the use of language to discuss or describe itself.

Directly connected to communication is *dialogue*, seen as the prototipical form of language use and communicative exchange [13, 14]. Dialogue is a communicative process which requires two or more interlocutors, who coherently transmit pieces of information in one or more dialogue turns. Dialogue can use different modality, both for input and output encoding: verbal (written, or spoken language, or both), non-verbal (gestures, facial expressions), multimodal (both verbal and non-verbal), multimedial (audio, video, pictures). The introduction of dialogue in machines can be explained considering one of the hypothetical cause for language origin. According to the practical glottogonic hypothesis, the language origin has functional reasons, as people started using it to facilitate the performing of practical tasks (i.e. hunting), whose organisation could be in this way made more efficient. Whether language was originated because of practical needs or for creative purposes (or even because of love, as suggested by Jean-Jacques Rousseau), it cannot be denied that the research on dialogue systems started for practical reasons. In fact, after the development of dialogue systems based only on written text, such as ELIZA [176], the task of building the first spoken dialogue system was assigned to the DARPA Project (Defence Advanced Research Projects Agency) around 1977, whose aim is to develop technologies for military use. More concretely, the first systems were used in the telecom industry, providing new telephonic services, such as automated agenda, and travel services.

Concerning the architectural structure of such systems, similarly to the two aforementioned communicative models, conversational agents are made up of different modules (Figure 1.1), each bearing specific functions:

Automatic Speech Recogniser In case of a spoken dialogue system, where the verbal interaction uses the spoken language, the voice input is processed by the Automatic Speech Recogniser (ASR), which returns the hypothetical transcription with a certain confidence by using acoustic models, or language models (grammars).

Natural Language Understanding The transcription outputted from the previous module is processed by the Natural Language Understanding (NLU) module, where the intent or the meaning of the input is recognised. This uses rule-based grammars or statistics-based models. The output of this module consists of a formal representation of the user intentions.

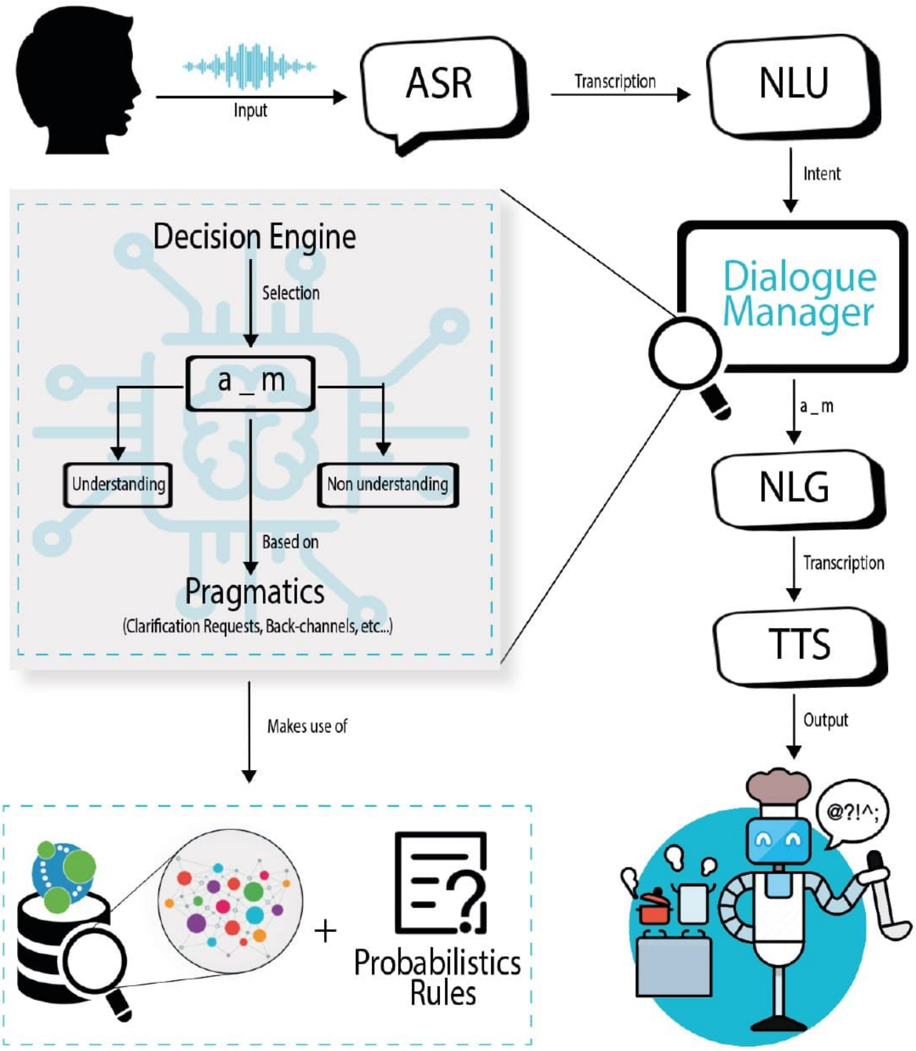


Figure 1.1: Dialogue System's Architecture

Dialogue Manager Represented as the decision engine of the dialogue system, the Dialogue Manager (DM) has the task of mapping the abstract semantic/logical form of the speaker's input to the corresponding output, i.e. the response action which better suits the one received in input (a_m in Figure 1.1), represented as actions which can reflect the degree of understating by the agent. In this decision-taking engine, all the formal characteristics of the dialogue are statistically, or deterministically modelled; hybrid approaches are also possible, as for OpenDial [98]. In other terms, this module deals with the formal characteristics of the dialogue that are studied in the field of pragmatics. These characteristics can be derived from external resources, such as by combining domain representations in a graph database and probabilistic rules (Chapter 6).

Natural Language Generation The previously outputted abstract form of the response action is linguistically processed in the Natural Language Generation (NLG) module, which also makes use of grammar, template structures and rules. The output returned corresponds to a textual action.

Speech Synthesis The textual form of the selected response action is physically reproduced by the Text-to-Speech (TTS) module, whose output is a speech signal. The final product could be , for example, a robot acknowledging the user of its level of understanding by using spoken language (i.e., *Ok, I'm going to mix flour with sugar*).

To train each module, different tools and resources can be adopted, as described in the next section.

1.1.1 Tools and Resources for Dialogue Systems Development

For the development of such systems, different approaches, data and tools can be used. For instance, as far as corpus-driven dialogue systems are concerned, there is a vast amount of data documenting human dialogues. Furthermore, annotation standards, annotation platforms, or tools for extracting different kinds of signals can also be adopted in the dialogue development framework. Here, we start focusing on applications which can be specifically used in the design of a dialogue system. In particular, we present some tools which are being used for the linguistic and paralinguistic development of conversational agents. For these purposes, in the next sections, the use of some sources are described, especially as far as input processor, dialogue modelling, and multimodal alignment are concerned.

1.1.1.1 Input Processor

By input processor, we mean here the pre-processing of speech data, on the basis of which the recognition of specific signals from the audio is modelled and defined. In fact, speech corpora can be used to extract prosodic profiles connected to communicative strategies, in order to train the system to consequently recognise them or use them in specific situations. For this purpose, the web service WebMAUS[3] can be used to fulfil specific phonetic requirements. The Munich AUtomatic Segmentation (MAUS) system [139, 82] is a multilingual tool used to transcribe audio inputs and align transcription to the spectrogram, returning as a result a TextGrid file[4]. Beside the graphic transcription, which can be provided or can be left to the integrated ASR

[3] https://clarin.phonetik.uni-muenchen.de/BASWebServices/interface

[4] A TextGrid file is a text file used for labelling segments of an audio file. It is used in Praat to show the labels aligned with the audio segments.

(Automatic Speech Recogniser), the tool also provides the phonetic one in SAMPA for each word and each phone, as in Figure 1.2. It also provides related services, such as TTS (Text-to-Speech), syllabification, and chunking. By using the resulting files, particular phonetic features, which can be associated to the semantics of linguistic intents, can be extracted, such as pitch and intensity. For the manual or automatic extraction, the Praat program [19] can be adopted. Furthermore, the obtained data can also be used to outline sociolinguistic profiling of speakers by extracting pieces of information connected to the openness of vowels and other articulative peculiarities, as in [48]. In the next section, this aspect will be highlighted with regard to the use of annotated spoken corpora with regional varieties, such as CLIPS [137].

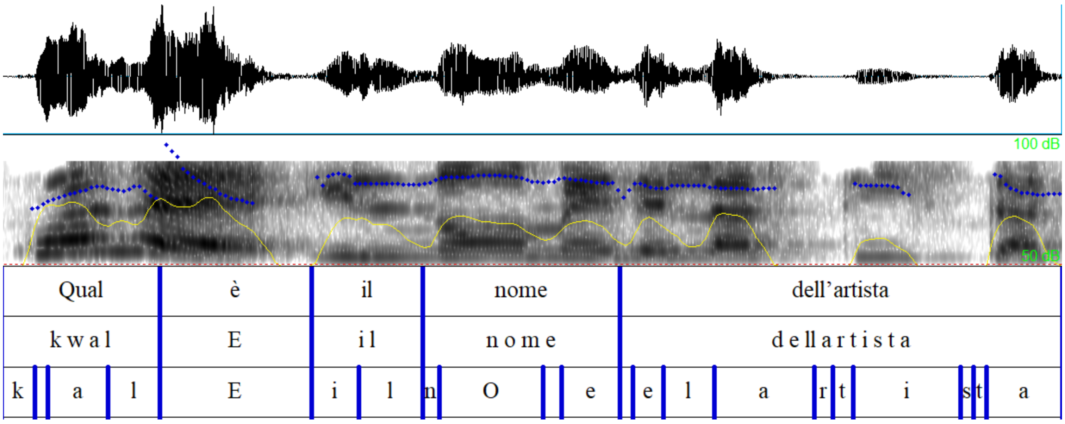


Figure 1.2: Resulting TextGrid file of a MAUS forced Alignment in Praat; the annotation levels correspond to i) ortographic transcription (It. *Qual è il nome dell'artista?* En. *What is the name of the artist?*), ii) phonological transcription to the word level, iii) phonological transcritpion to the phone level

1.1.1.2 Dialogue Modelling

Dialogue Modelling refers to the design of the dialogic exchange as far as intents definition and output mapping are concerned. Strictly connected to dialogue modelling is the definition of the communicative strategies arising in conversation, among which the turn-taking organisation [133], or the use of clarification requests in non-understanding or conflicting scenarios are to be mentioned. For the semantic and pragmatic design of dialogues, different sources can be exploited. Among various techniques, the use of SRGS (Speech Recognition Grammar Specification)[5] [77] is mostly preferred to assure the categorisations of possible intents in a target-oriented dialogue system, by means of the description of each possible structure that can be uttered to express a particular concept. The use of grammars is especially suitable for commercial systems, whose domains can be deterministically better defined, avoiding

[5] Speech Recognition Grammar Specification Version 1.0: https://www.w3.org/TR/speech-grammar/

relying on error-prone machine learning algorithms. These grammars can be automatically extended, as far as lexical variability and inflectional morphology is concerned [53], making use of semantic networks such as ItalWordNet [131] and POS-tagging tools like Tree-Tagger [142].

The language model to be used for conversational purposes can be enriched with pragmatic information. For a more natural and efficient communication, the system is expected to be able to understand speech acts and, therefore, user intentions. Furthermore, in case of non-understanding, it also has to be able to signal the problem efficiently in order to solve it (for further details, see Chapter 1.1.2). For this purpose, different annotation techniques have been employed to model and understand pragmatic phenomena. For instance, the Dialogue Act Mark-up Language (DiAML) could be used. For 'dialogue act' it is intended, as reported in ISO standard 24617-2,

> *"[...] a stretch of communicative activity of a dialogue participant, interpreted as having a certain communicative function and a semantic content, and which may additionally have certain functional dependence relations, rhetorical relations, and feedback dependence relations".*

Not only is it suitable to annotate the type of intent performed, but it is also effective to specify further information: i) whether the user intent was merely dependent on the action motivating the dialogue itself; ii) whether it was a feedback to the previous turn (auto- and allo-feedback); iii) if it was signalling the turn-giving or turn-taking action; iv) opening, closing or structuring the conversation; v) in case of social obligations adjacency pairs [27]. The specification of the performed act is indeed useful to improve the disambiguation and thus the understanding. For instance, asking for more information or for clarifications is important to ensure that the interlocutors are on the same page, namely that a common ground is established.

Besides rule-based approaches, which can make use of grammars, we can use corpora for the statistical extraction of knowledge. Data analysis can be both corpus-based and corpus-driven: on the one hand, a given corpus can help to confirm or refute a pre-existing theoretical construct (corpus-based), on the other hand a corpus can be used to generalise rules (corpus-driven). For modelling conversational interactions, spoken corpora are useful to capture domain-dependent semantic aspects and the pragmatic characteristics arising from dialogue. Therefore, a corpus-driven approach is usually adopted. To achieve such aims, the construction of tools like SPOKES is truly interesting. SPOKES - currently available in Polish and English – is an online service for conversational corpus data search and exploration [116]. By exploring this corpus, information concerning the strategies used in conversation can be extracted to be modelled in a language model. Providing pragmatic annotation in such tools could be an advisable goal to make it better applicable in the development of conversational agents. As it will be further underlined in this work, pragmatic studies are, therefore, of primary importance when dialogue is involved. As for the current availability of spoken corpora for Italian, some of them are summarised in Table 1.1.

Corpus	**Annotation**
AN.ANA.S._MT [6]	syntactic information
Corpus AVIP-API [7]	orthographic transcription
CLIPS[8]	segmental information
EXMARaLDA Demo Corpus [9]	suprasegmental information, accentuation/stress marking
SpIt-MDb [10]	acustic, phonetic, phonological, and lexical information

Table 1.1: Italian Spoken Corpora

In particular, CLIPS [137] contains dialogues from speakers coming from 15 different regional varieties of Italian. This could be useful to train a system to recognise the geographical origin of the speaker for profiling purposes. Among others, we mention AN.ANA.S [168] which contains syntactic annotations and whose information could be used for training the system to recognise syntactic structures and disambiguate semantic usages.

In a multimodal perspective, speech and gestures corpora are a further asset in the use of data for training dialogue systems. In particular, deictic information or ellipses can be recovered by the listener via the the interpretation of gestures. An explanatory example is drawn from the SaGa Corpus [101]. The SaGA corpus consists of 280 minutes of video material containing 4961 iconic/deictic gestures, approximately 1000 discourse gestures, and 39,435 words. The annotation comprises gesture segmentation and classification (iconics, deictics, beats), gestural representation techniques (e.g., drawing, placing), morphological gesture features (e.g., hand shape, hand position, palm orientation, movement features), transcription of spoken words and dialogue context information, based on DAMSL dialogue acts, information focus, and

[6] AN.ANA.S._MT Corpus. Archived at the University of Salerno. Published in 2010. `http://www.parlaritaliano.it/index.php/it/corpora-di-parlato/716-corpus-ananas-multilingue-ananasmt`

[7] Corpus AVIP-API. Archivio del Parlato Italiano. Archived at the University of Salerno. Published in 2003. `http://www.parlaritaliano.it/api/`

[8] CLIPS Corpus. Archived at the University of Naples 'Federico II'. Published in 2005. `http://www.clips.unina.it/it/`

[9] "EXMARaLDA Demo Corpus 1.0." Archived in Hamburger Zentrum für Sprachkorpora. Publication date 2007-11-08.
`http://hdl.handle.net/11022/0000-0000-4F70-A.`

[10] SpIt-MDb Corpus (Spoken Italian - Multilevel Database). Archived at the University of Salerno. Published in 2006.
`http://www.parlaritaliano.it/index.php/it/corpora-di-parlato/644-spit-mdb-spoken-italian-multilevel-database`

thematisation [101]. This corpus will be the starting point of the analysis presented in this work (Chapter 3), thanks to the access provided by the Social Cognitive Systems Group (CITEC)[11] led by Professor Stefan Kopp during the six-month period spent at the University of Bielefeld.

The use of multimodal corpora is also particularly interesting when considering that identical utterances can take on different meanings according to not only the prosodic structure of the message being conveyed but also according to the gestures or facial expressions we use while uttering it. The collection of multimodal corpora is, therefore, configurable as a necessity. For the Italian language, there are not a lot of data sources, besides language learning (L2) collections, such as the TAITO-project. Nevertheless, a multimodal and multi-party corpus for the Italian language, specifically applied in the cultural heritage domain, has been collected for the CHROME project (*Cultural Heritage Resources Orienting Multimodal Experiences*)[12], whose aim is to define a methodology of collecting, analysing and modelling multimodal data in designing virtual agents serving in museums [111, 43].

1.1.1.3 Multimodal Alignment

As a completion of the description of communicative models, theoretical and computational (i.e. FANTASIA [110], see also Chapter 6), we also refer here to multimodal alignment. The module responsible for the fusion of different channels of intents' communication – spoken language and paralinguistic features, specifically gestures and prosodic profiles – can rely on data synchronised with a tool like ELAN, before being learned through probabilistic rules or machine learning algorithms. ELAN is a tool designed to annotate audio and video files [178]. In ELAN's tiers, TextGrids, which are, for instance, obtained with WebMAUS, can be imported and overlapped to other pragmatic and paralinguistic information. The fusion of different annotation levels can be used to process both the understanding and generation processes. For instance, this tool is being used within the CHROME project to specifically model the way the gatekeeper would communicate cultural contents (Figure 1.3). After having recorded authentic tour guides, video and audio files have been synchronised in ELAN, where expert annotators marked linguistic and paralinguistic phenomena [111]. In addition to that, the postures, gestures and facial expressions of listeners are annotated to capture their aptitude towards the content being conveyed. Fusing different channels of communication together in the modelling phase will result in a virtual tourist guide able to communicate as naturally as human ones, capable of adapting their communicative strategies

[11] https://scs.techfak.uni-bielefeld.de/

[12] www.chrome.unina.it

to the type of interlocutor. In addition to ELAN, pragmatic phenomena can also be manually annotated using tools such as EXMARaLDA [143], a system for the computer-assisted creation and analysis of spoken language corpora.

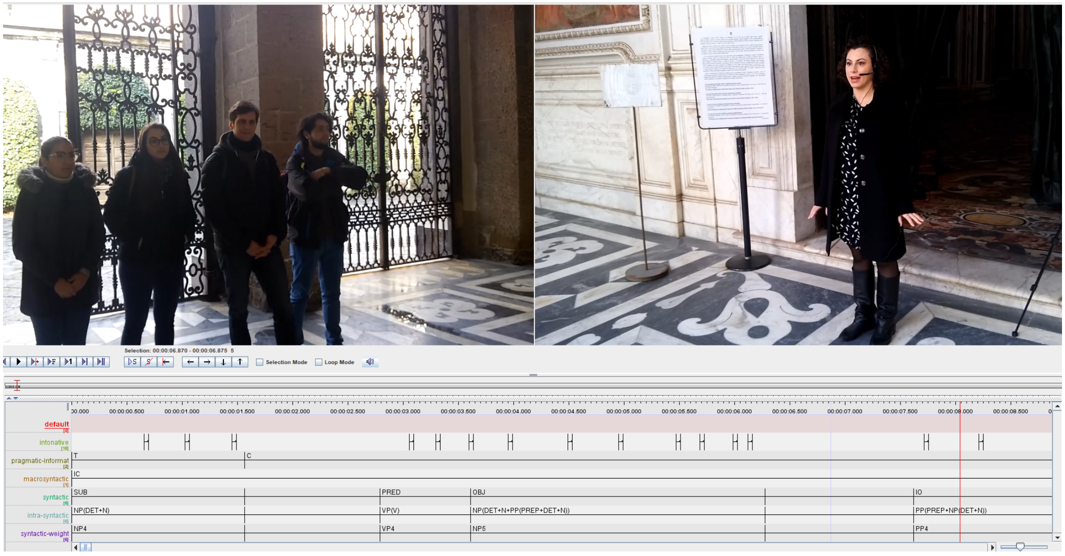


Figure 1.3: Example of the multimodal annotation of the CHROME corpus via ELAN

1.1.2 Challenges in Conversational Agents

Beside the structure a dialogue system can have, and beside the training techniques and materials that can be used, other aspects are important to take into account when dealing with conversational agents. In fact, conversing is a linguistic task that conversational agents still need to explore in their totality. Conversing involves different aspects. First of all, it requires *Understanding*. Understanding means a) processing what the interlocutor is saying in terms of words identification (i.e., *Please close the window*); b) processing the meaning of the words communicated (i.e. Request(subject: close (object: window))); c) processing the speaker meaning, that is what a speaker intends to convey through the message communicated. The interpretation of the speaker meaning follows four steps: i) given that the interlocutors share knowledge, inferences are encoded in the message considering the common ground; ii) the starting point of the intent interpretation is the literal meaning of the message communicated; iii) the result of this process is the understanding of the intention of the speaker; iv) to fully understand what the agent A intends, it is important to also identify the behavioural game A is implicitly or explicitly referring to [11, p 158]. For behavioural games, Bara [11] intended the stereotypical interactional schemes interlocutors think they are into. Direct and conventional indirect speech acts are simple communication acts when they immediately refer to a behavioural game, whereas non-conventional indirect speech acts are complex communication acts when they do not immediately refer to a behavioural game [76, p. 295]. The interactional schemes of behavioural games are controlled by the meta-level of analysis represented by the communicative competence. In fact, the second aspect of conversation that one should pay attention to when modelling dialogue systems is the communicative competence. Hymes [78] described the communicative competence of speakers as the ability to know *when to speak, when not, what to talk or not talk about, with whom, when, where, and in what manner*. Strictly connected to this concept is the one of *interactional intelligence*. With *interactional intelligence*, we mean the ability to recognise intentions, beliefs, and aptitude towards the dialogic exchange and the ability to respond appropriately [94, 30]. In fact, conversing involves being capable of managing the interaction. This skill is fulfilled when a) interlocutors have an internal representation of the domain (Communal Common Ground and Specialised Common Ground[13] [36]); b) interlocutors are capable of recognising new information (Personal Common Ground and Local Common Ground [36]); c) interlocutors are capable of selecting the corresponding most appropriate action (verbal or not verbal), according to the illocutionary or perlocutionary force of the input and to the shared knowledge, and are capable of framing the interaction in a specific behavioural game. One important part of this competence is, therefore, to have access to the domain knowledge but also to the context. For example, if the agent A asks *Please, open the window*, the agent B, in order to understand and respond properly, not only needs to know what a window is or what A meant with the request, but B also needs to know that there is a window, which is currently open, and that can be closed. In fact, if A knows that some aspects of the context are not accessible to B, A must specify this information with the aim of building a common ground, so that A can be assured that B can understand the request and respond appropriately. If this information is missing, other communicative tools can be used to make up for this lack. All these phenomena are studied in pragmatics.

The wide success and the current spread of conversational agents are indeed shedding a new light on conversation analysis and on the pragmatic structure of dialogue. Specific interest is drawn by the study of the automation of pragmatic phenomena which are common in human-human interactions [33, 28, 5, 150, 75]. In the pragmatic analysis of conversation, the starting point is to define dialogues as joint activities, for which the joined goals of the interlocutors and their role in a particular interaction must be identified in order to reach the conversational targets [103]. Each utterance we produce in a spoken interaction is the result of an act of cooperation. As pointed out by scholars such

[13] For the definition of different types of Common Ground, see Chapter 2

as [37], to pursue the aim of succeeding in their joint activity, the interlocutors need to *ground* what is being communicated. In conversation analysis, *grounding* refers to the act of establishing that what we intend to say (or what has been said) can be well understood (or has been well understood) [38]. To establish a *common ground*, different strategies, such as linguistic or paralinguistic feedback analysis [163], can be exploited. From the linguistic point of view, dialogue efficiency can rely on the analysis of communicative feedback, whose relevance was pointed up by Allwood et al. [4] and which continues to be considered as an important characteristics in dialogue modelling [32].

In this work, specific attention is dedicated to computational pragmatics, which aims at simulating the encoding and decoding mental processes of interlocutors when conversing. More in depth, the pragmatic tools represented by clarification requests used in specific inconsistent contexts are studied. Since inconsistencies are mentally reconstructed based on the relationships between pieces of information part of the knowledge structure shared within a community (i.e. Communal Common Ground), and since the interpretation of the internal state of the agent using such tools is an exercise in the theory of mind, details concerning cognitive pragmatics are needed as introductory concepts.

1.2 Cognitive pragmatics

Differently from the general linguistic concept of pragmatics, cognitive pragmatics aims at investigating what happens in participants' minds [76, p. 281]. Cognitive processes involved in communication are based on three fundamental processes:

Cooperation Since Grice [69], the importance of cooperation in a successful conversation was pointed out. The cooperation model of communication was described in Tomasello [161] (Figure 1.4): the communicator C has individual goals, such as goals and values pursued in their life. If for any reason, C feels that the recipient R can be of any help in the achievement of some goals, C will produce specific acts which will bring R to do something, know something, or share something. This is represented by C's social intention, which is expressed through communication. Therefore, a communication act (verbal or not verbal) is mutually manifested in the joint attentional frame. The C's communicative intention is consequently shared. C can also draw R's attention to some referential situation in the external world (referential intention) designed to lead R to infer social intentions via processes of cooperative reasoning [76, p. 282]. On the other hand, R attempt to firstly identify the referent, typically within the space of the common ground, and secondly to infer the social intention, also by relating it to the common ground. Then,

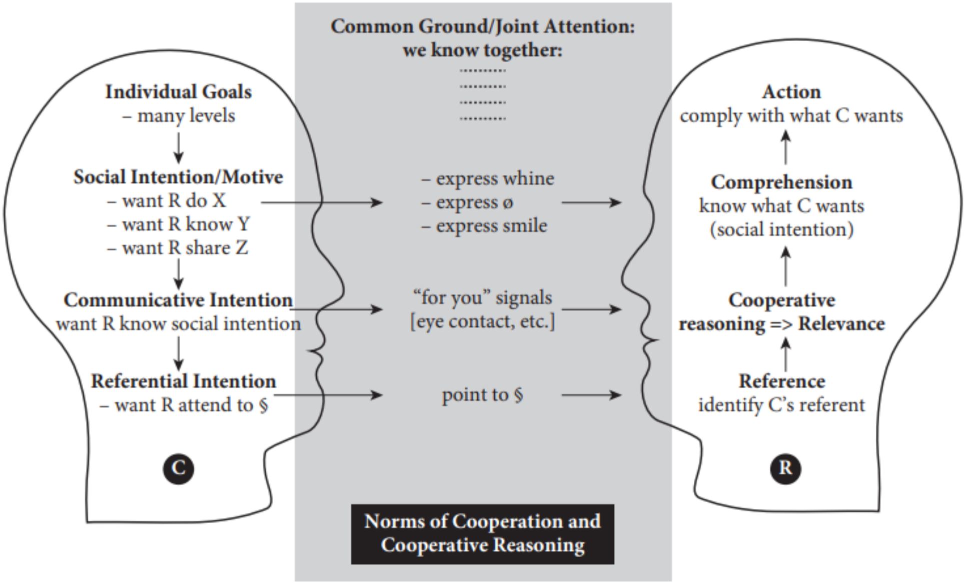


Figure 1.4: Summary of cooperative model of human communication (C = communicator; R = recipient) [161]

assuming that R understands C's social intention, R can decide whether or not to cooperate as expected [161, 76].

Sharedness Cooperation is not sufficient to allow communication. Human beings are also capable and willing to share mental states (i.e., emotions, beliefs, intentions, and desires) [76, p. 283]. Among mental states, *beliefs* are to be listed. Beliefs can be of three types: i) individual - personal beliefs agents can have for themselves or in representing others, without an existing connection between the agents; ii) common - individual beliefs agents mutually share in a given context; iii) shared - beliefs common to all participants engaged in a conversation, and that each participant knows it is possessed by all other participants. Sharedness depends on another important factor defined *common ground* by Clark [37, p. 93], that is *the sum of knowledge, beliefs, and suppositions that two or more people share*[6]. Of course, shared beliefs also have subjective features, as no one can ever be certain that a particular belief is shared among all interlocutors. Cooperation and help identify the sharedness of mental states.

Communicative intention No communication can ever occur if there is no intention to communicate something. Communicative intention has been defined by Grice [69] as the intention to communicate something, plus the intention that that intention is recognised as such.

[6] More details on Common Ground will be given in Chapter 2.

In dealing with mental processes involved in communication, cognitive pragmatics does not only investigate understanding and generation processes in standard situations, i.e., in situations which follow the rules of specific behavioural or communication games. In fact, other non-standard types of communication are taken into account: i) *non-expressive interaction*, when an utterance is used not to express a mental state; ii) *exploitation*, the use of a communication rule for producing communicative effects different from the ones typically associated to that rule (i.e., irony); iii) *deception*, when a not possessed mental state is conveyed; iv) *failure*, when the desired communicative effect is not achieved [76, p. 291].

A particular case of non-standard communication can also be represented by *conflicting representations*. Conflicting representations take place when there is a discrepancy between what is communicated and what is believed by the agent. In these scenarios, default rules are violated and more sophisticated mental representations are required [76, p. 294]. In this work, I will concentrate on a particular example of conflicting representation, constituted by common ground inconsistencies, for which a grounded information clashes with a new communicated one. This representation thus needs specific mental representations in human-machine as well as in human-human interaction and linguistic strategies to account for them in a cooperative perspective in communication.

1.2.1 Theory of Mind

As linguistic strategies adopted in conflicting scenarios are transpositions of mental states concerned with epistemic bias (Chapter 2), mental representations are important criteria in cognitive pragmatics as well as in this work. In such a context, mentalising becomes a theoretical asset. *Mentalising* is defined as the process of making inferences about the mental states of other agents [60]. A prerequisite of mentalising is that agents have a *theory of mind*. This is defined as the ability *to infer the full range of mental states (beliefs, desires, intentions, imagination, emotions, etc.) that cause action. In brief, having a theory of mind is to be able to reflect on the contents of one's own and other's minds* [12, p. 174]. This means that agents are aware of possessing mental states and can also attribute mental states to other agents. How exactly the attribution and inference processes work is still disputed. These processes also depend on the type of mental state that is to be inferred. Mental states are generally thought to fall into two categories: i) *propositional attitudes*, which have a content, in that they refer or are about something, as for beliefs, knowledge, intentions, fears, and doubts; ii) *phenomenal states*, which correspond to the quality of experiencing things, like pain, thirst, sadness, uncertainty, and colours [32]. Hybrids of the two types of mental states are also possible [44].

Theory of mind is often studied in evolutionary terms. For instance, a common subject of its testing is whether children have a theory of mind, or, in other words, whether they are capable of inferring other people's mental states. One way to test it is through the false belief task [132]. This task

> *" [...] involved a character, Maxi [a puppet], who places some chocolate in a particular location and then leaves the room; in his absence the chocolate is then moved to another location. The child is then asked where Maxi will look for the chocolate on his return. In order to succeed in this task, the child must understand that Maxi still thinks that the chocolate is where he left it — the child must understand that Maxi has a false belief, in fact"* [34, p. 2].

In the words of this work, the child should, therefore, be able to understand the conflict existing between a presupposition and a new contextual evidence, which is, in this case, not yet part of Maxi's common ground.

As far as conversation analysis is concerned, recognising what the interlocutors know and being able of adjusting actions and understandings accordingly are central topics. These abilities are directly connected to the theory of mind. Inferring another agent's mental state is a prerequisite for different speech acts, such as lying, beliefs manipulation, and others. The inference of mental states corresponding to propositional attitudes, for instance, guides the interaction and can determine the use of specific conversational feedback. One example is the use of the corrective feedback in case of *epistemic bias*, i.e., the use of clarification requests in the form of polar questions when one agent believes that a known informational item is true or false compared to the one grounded by the interlocutor (Chapter 2). Each person, indeed, needs to know what other agents know and the reliability of their claims. In other words, speakers need to know the *epistemic status* of others. Sperber et al. [153] called this need *epistemic vigilance*. This necessity guides actions on what is being asserted. For instance, epistemic vigilance determines the check for pre-conditions and post-conditions of actions against previously grounded informational items, before making these actions part of the common ground (Chapters 5 and 6).

1.2.2 Cognitive Pragmatics and Dialogue Systems

When pragmatics is applied to dialogue modelling, we talk about computational pragmatics, especially as far as the development of dialogue systems is concerned. In fact, computational pragmatics mostly deals with corpus data, context models, and algorithms for context-dependent utterance generation and interpretation [76, p. 326]. Nevertheless, conversational agents should be able not only to process local but also global structures of dialogues [1]. Whereas local structures are involved with linguistic rules (i.e., speech acts, turn-taking, etc.), which can be derived from corpus analysis, global structures refer to the conversation flow, that is the dialogue's action plan and how this is mutually known by dialogue participants (i.e., opening, closing, etc.). Cognitive pragmatics looks at these global structures derived from behavioural games, which in turn derive from grounding processes [11]. Different authors started including these processes in their dialogue systems architectures, especially as far as evaluating and updating common ground with their human partner, which is also the main topic of this work. For instance, Roque and Traum [130] have developed a dialogue system that tracks grounded information in the previous conversation. As a consequence, the dialogue system is capable of selecting its utterances using different types of evidence of the user's understanding (i.e., whether the dialogue system has just submitted material or the user has also acknowledged it, repeated it back, or even used it in a subsequent utterance) [109].

Using grounding strategies in conversational agents brought to interesting implementations. One aspect which has not yet been investigated is concerned with the mechanisms of grounding between humans and dialogue systems. Experimental investigations have mostly studied *how users evaluate the interaction, instead of studying interaction mechanisms* [109, p. 3]. For instance, Roque and Traum [130] performed a user study in which subjects interacted with their system and rated how much they felt the system understood them, put effort into understanding them, and gave appropriate responses. Conversely, what most studies do not ask is how a specific dialogue principle, such as the use of a particular type of request, is used by a system to affect user behaviours. Therefore, to learn more about human–machine dialogues mechanisms, it is important to turn to more basic experimental researches [109], like the one presented in this work.

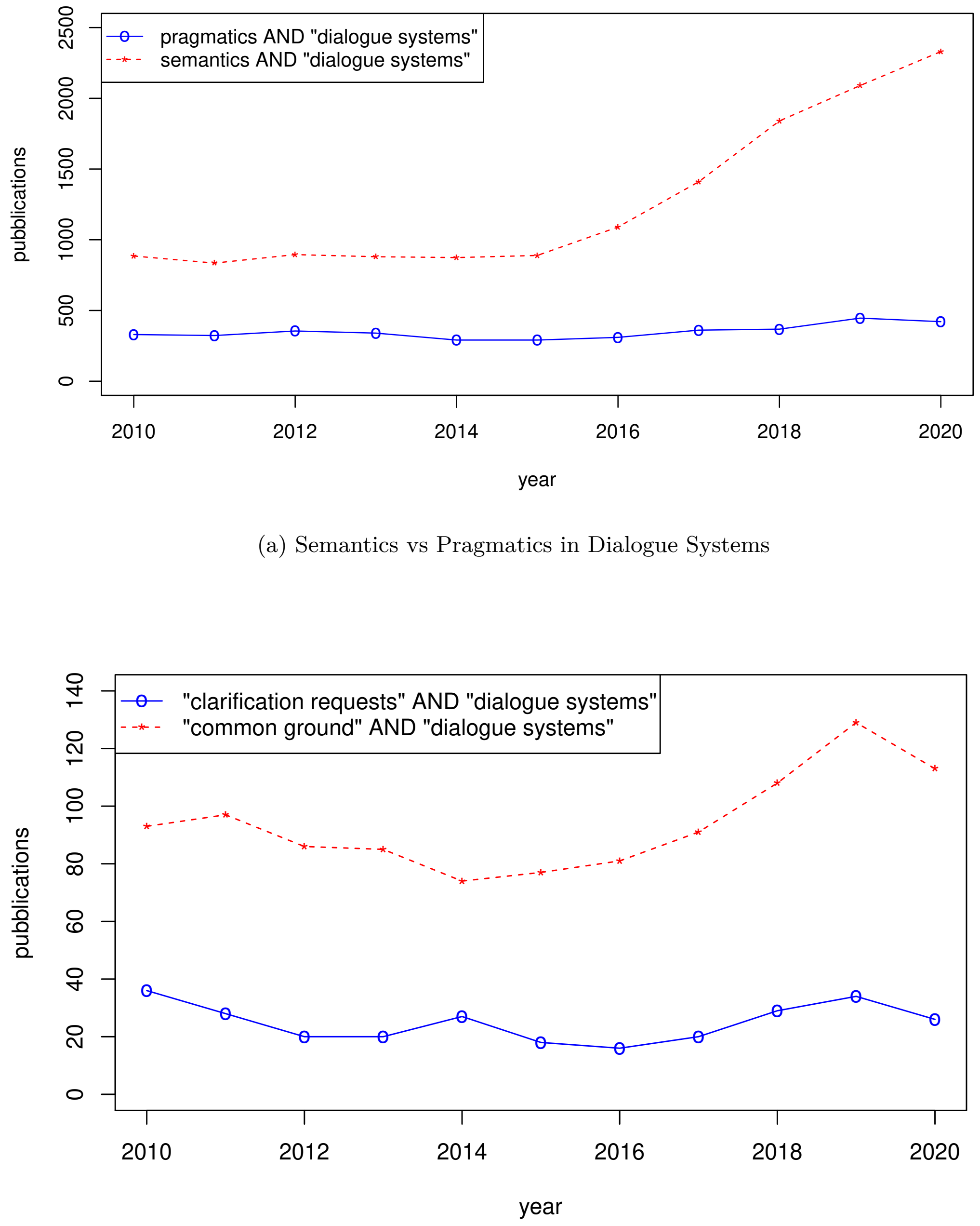
1.3 Motivations and objectives

The general objective of this dissertation is to investigate how inconsistencies in the knowledge stored in the Common Ground can be efficiently signaled by a machine conversing with human users in order to enable them to recognise the current internal state of the system and recover errors. Among various typologies of dialogue systems, specifically the ones that do not exclusively deal with the understanding of single separated commands, a particular type of system is to take into account. We call it *User Guided Task Application.* This builds the information incrementally to learn a task. In this context, the ability to check the consistency between the information received becomes indispensable. Therefore, when facing consistency problems, the formulation of clarification requests, with the consequent application of language generation and speech synthesis algorithms, becomes a matter of research. The specific objective of this work is to highlight the nature of particular clarification requests, those in the form of polar questions, which not only are used to express a conceptual and/or informative inconsistency toward a presupposition, but can also denote the kind of problem and the mental state of the speaker toward the problem itself.

The importance of focusing on such topics reflects the need to bridge the gap in the study and development of dialogue systems left by the lack of insights into the application of pragmatics to conversational agents. Although pragmatics is the level of language analysis strongly depending on dialogue, its computational application is mainly focused on the study and identification of speech acts [92]. On the other hand, as also shown in Figure 1.5a, in the last ten years, semantics has been a more investigated topic within the dialogue systems field with respect to pragmatics, especially as far as the understanding of the correct recognition of the received intent was concerned. In more detail, in the field of pragmatics, in the last ten years (Figure 1.5b), the research on Common Ground has seen a thriving impulse, as shown in the publications on dialogue systems. Despite the fact that Clarification Requests are one of the grounding tools used by interlocutors while conversing, their study and application in dialogue systems have not yet seen a boost. All in all, a more in-depth analysis of pragmatic phenomena related to Common Ground construction and consistency checks in human-machine interaction, such as the use of Clarification Requests, appears to be a missing spot in the research on dialogue system, and whose necessity needs to be confirmed in terms of efficiency increase with the support of the here presented study.

This dissertation deals both with linguistic and computational aspects of the following processes:

Common Ground Inconsistencies Given a domain D, we define sequential actions as a number of different d, referred to as various *domain-related commands*. As a result, the domain can be represented as $D = \{d\}$. Each d is characterised by a number of states $S = [s]$ representing pre-conditions $s\_pre = [\,]$ and post-conditions $s\_post = [\,]$ depending on the type of action. D is inconsistent when d presents at least one s, whose s_pre and/or s_post are incompatible with respect to another d, as they cannot co-exist. When this conflict takes place an inconsistency occurs. This conflict can depend on



(a) Semantics vs Pragmatics in Dialogue Systems

(b) Clarification Requests and Common Ground in Dialogue Systems

Figure 1.5: Google Scholar's plotted results about publications on dialogue systems applying semantics versus pragmatics; as far as pragmatic phenomena are concerned, publications on clarification requests and common ground are plotted [Retrieved on 30/11/2020]

i) a *s_pre* which is incompatible with the rules of the Communal Common Ground (i.e., *cut the milk*) ii) the incompatibility of *s_pre* of the current *d* with *s_post* resulting from a preceding *d*, saved in the set of shared knowledge - the Personal Common Ground. Although both Common Ground Inconsistencies can cause corrective feedback, only the second type is linked to the adoption of Clarification Requests.

Clarification Requests Among the corrective feedback that can be used in conversation to solve various problems, ranging from acoustic to semantic ambiguities, and from syntactic to pragmatic issues (i.e., Common Ground Inconsistencies), Clarification Requests are here considered. Being a *grounding* tool, Clarification Requests are an important pragmatic device adopted to establish and maintain successful communication [37, 3]. This peculiar feedback presents different functions and forms. In this work, Common Ground Clarification Requests in the form of Polar Questions will be analysed. This specific type of clarification dialogue is initiated when Common Ground Inconsistencies take place, namely when conflicts between an agent's presupposition and a given evidence occur.

Argumentation-based Dialogue Argumentation-based Dialogue refers to the modelling of the verbal interaction aimed at the resolution of conflicts of opinions via the adoption of specific strategies. This field of study consists of a variety of different approaches and individual systems, with few unifying accounts or general frameworks [120]. Among the types of Argumentation-based Dialogue, we mention the deliberation dialogue, whose application is aimed at reaching a decision or at establishing a course of action. This type of dialogue can, indeed, be compared to the map-tasks, analysed in this work, or to the type of dialogue used for the experiment illustrated in Chapter 5, where the course of action is represented by the set of *domain-related commands* $D = \{d\}$, we previously referred to.

Chapter 2

Grounding Theory of Communication

In this Chapter, the concept of common ground will be introduced and, particularly, the importance of the grounding process in conversation. This process is an important starting point to understand the phenomena described in this work, concerned with conflicts occurring between information grounded in conversation. As Ginzburg [63] argues, interlocutors are constantly monitoring each other to seek evidence about the understanding of the last utterance by the interlocutor. If the exchange succeeds, the information is grounded and becomes part of the shared information of the speakers; if it is not, clarifying previous information is required in order to fully achieve their informational goals. Such exchanges are called clarification dialogues, that is, when i) one interlocutor is posing questions to the other who gives answers, ii) when the dialogue clarifies some concepts in questioner's mind, either by asking new information or by asking clarification or explanation about the given information [45]. Requests for clarification and their classification will be deeply discussed in the last sections, with particular attention to Common Ground clarification requests and their most common syntactic form, i.e., polar questions.

2.1 Grounding: From Accumulation to Argumentation

As Stalnaker explained, *when speakers speak, they presuppose things and what they presuppose guides both what they choose to say and how they intend what they say to be interpreted. To presuppose something means to take it for granted as background information – as common ground among the participants during their conversation* [155, p. 701]. In fact, communication is a joint activity in which two speakers must share information or, in other words, have a common ground, that is mutual knowledge, mutual beliefs, and mutual assumptions, in order to understand each other [38]. To coordinate on this process, speakers need to update their common ground from time to time. Common ground, as Clark [36] acknowledged, can be of four main types: personal, local, communal and specialised. *Personal Common Ground* (PCG) is established collecting information over time through communicative exchanges with an interlocutor and it can be considered as a record of shared experiences with that person. A part of PCG is *Local Common Ground* that is tied to a piece of information obtained from a single exchange with an unknown or known interlocutor. According to Clark [36], information of this type can be, for instance, the opening hours for a shop, train timetables, and so on. With *Communal Common Ground* (CCG), it is intended an amount of information shared with people that belong to the same community, that is to say, people that share general knowledge, knowledge about social background, education (schools attended, levels of education attained), religion, nationality, and language(s). Within a larger community, a smaller one can be found: *Specialised Common Ground* pertains to those people that share particular areas of expertise about some domain of knowledge, such as colleagues, friends, or acquaintances, and it is marked by specialised vocabulary of that specific domain, such as medicine, law, and so on. For the purposes of this work, only PCG and CCG are going to be considered.

All collective actions are built on common ground and its accumulation. In communication, common ground cannot be properly updated without a process called *grounding*. Grounding is an important part of communication and, for this reason, it is fundamental to understand how it works. Following Clark and Shaefer's theory [39], we suppose that a participant, called *contributor*, contributes to a conversation with their partner. According to their proposal, making such a contribution requires two things: i) the contributor specifies the content of their contribution ii) the partners register the contribution content (content specification). The second requirement is connected to what is called *grounding criterion*, according to which *the contributor and the partners mutually believe that the partners have understood what the contributor meant, to a criterion sufficient for current purposes* [39]. This grounding process enables common ground to accumulate information in an orderly fashion. Accumulation and grounding, therefore, work together to create a unit of conversation that Clark and Shaefer [39] call a *contribution*. In a conversation, as mentioned before, contributions begin with a contributor A (here female) presenting an utterance to her partner B (here male) that must register and understand it. A does not know if she has succeeded in her intention unless the hearer provides evidence for his understanding. A must believe that she has succeeded in communicating the utterance. That require that A and B mutually believe that B has understood it. More precisely, A must come to believe she and B

have satisfied the grounding criterion, and so must B [39]. Thus, contributing can be divided into two phases:

Presentation phase A presents utterance u for B to consider. She does so on the assumption that, if B gives evidence e, she can believe that he understands what she means by u.

Acceptance phase B accepts the utterance u by giving evidence e that he believes he understands what A means by u. He does so on the assumption that, once A registers that evidence, she will also believe that he understands.

In other words, A presents an action for B to consider, and B accepts that action as having been understood. If these two steps are done right, A and B will each believe they have arrived at the mutual belief that B understands what A meant by his action [39]. Grounding becomes most evident in the acceptance phase because after the presentation of utterance u by A, the partner B may believe A is in one of these states for all or part of u.

When we add something in a conversation, we are therefore looking for feedback, that is a *negative or positive evidence* that our partner has understood our contribution. Speakers can rely on feedback to find a possible negative evidence, an evidence that what we have said has been misunderstood or misheard. Feedback, according to Allwood [4], is a linguistic mechanism which enables interlocutors to exchange information. Feedback can refer to four different basic communicative functions: i) *contact*, whether the interlocutor is willing and able to continue the interaction; ii) *perception*, whether the interlocutor is willing and able to perceive the message; iii) *understanding*, whether the interlocutor is willing and able to understand the message; iv) *attitudinal reactions*, whether the interlocutor is willing and able to react and (adequately) respond to the message, specifically whether he/she accepts or rejects it. According to the author, linguistic feedback is usually composed by a short morpheme, repetition, head gesture, or body movements in combination with simple phonological, morphological and syntactic items [4]. Furthermore, communicative feedback tends be used as back-channels which do not usually adhere to the mechanism of turn-taking. In other words, feedback does not always occupy an entire turn, and it can be placed in parallel to it. The main types of positive evidences of understanding have been classified by Clark and Schaefer [39], from strongest to weakest: a) *display*, B displays verbatim all or part of A's presentation; b) *demonstration*, B demonstrates all or part of what he has understood A to mean; c) *acknowledgement*, B nods or uses fatic expressions (i.e., *uh huh*, *yeah*, etc.); d) *initiation of relevant next contribution*, *B* starts in on the next contribution that would be relevant at a level as high as the current one; e) *continued attention*, *B* shows that he is continuing to attend and therefore remains satisfied with *A*'s presentation.

According to Clark and Brennan [38], the first main form of positive evidence are *acknowledgements*, in particular back-channel responses that include continuers such as *uh, huh* or *yeah.* These are useful to signal that the utterance has been understood and that there is no need to initiate a repair in the next turn. Acknowledgements also include assessments (i.e., *gosh*, *really*) and they are usually produced without taking the turn. A second form of positive evidence is the initiation of the *relevant next turn*: suppose *A* is trying to ask *B* a question; if *B* understands it, the answer will be expected in the next turn. Questions and answers constitute adjacency pairs. In other words, once the first part of the adjacency pair is on the table, the second part is considered as conditionally relevant for the next turn. The third and most basic form of positive evidence is *continued attention* provided by an attentive listener. In conversation, people monitor their partner from time to time and immediately adapt to their feedback. If *A* utters something and notices that *B* was not paying attention, *A* could assume that *B* did not understand him. *B* must show that he is paying attention through eye gaze or communicative feedback. *A* can, therefore, use fatic expressions to understand if *B* is following, or she can elicit attentive listener feedback in *B*. On the other hand, *B* could want to show his attention by using such communicative feedback. Positive evidence of understanding, thus, is provided by communicative feedback and comes with attention that is unbroken or undisturbed [30, 31].

As argued by Clark and Schaefer [39], the strength of evidence that *B* has understood *A* depends on several factors, including the complexity of the presentation, how important the recognition is, and how close the interpretation has to be. The acceptance phase can be recursive, since *B*'s acceptation to *A*'s presentation needs to be accepted as well. They try to avoid infinite recursion in taking on acceptance phases by invoking the following *Strength of Evidence Principle*, which states that

> *"The participants expect that, if evidence $e0$ is needed for accepting presentation $u0$, and e1 for accepting presentation of $e0$, then e1 will be weaker than $e0$."* [163, p. 2]

The authors reported an example to better understand the principle: *A* presents a book presentation number, *f six two*, *B* accepts it by displaying it verbatim *f six two*; then *A* accepts the *B*'s acceptance by using a weaker evidence like *yes*. Lastly, *B* accepts the *A* evidence by proceeding to the next contribution. The traditional version of this principle exhorts speakers not to expend any more effort than they need to get their addressees to understand them with as little effort.

Grice [69] expressed this idea through two maxims: according to the maxim of *quantity*, the speaker must not make their contribution more informative than it is required, and, according to the maxim of *manner*, they must also be brief and avoid prolixity. However, the principle of least collaborative effort does not allow for grounding and, thus, it cannot represent what happens in real conversations. As claimed by Clark and Wilkes-Gibbs [40], there are three main problems with this principle: i) *time pressure*, speakers tend to limit the effort for planning an utterance, and this could lead to say improper utterances; ii) *errors* that a speaker makes during the uttering that need to be repaired; iii) *ignorance*, when a speaker does not know much about their interlocutor, they tend to generate improper utterances. Therefore, the authors refer to another principle, which states that *in conversation, the participants try to minimise their collaborative effort — the work that both do from the initiation of each contribution to its mutual acceptance* [40].

According to Clark and Brennan [38], there are two main factors that shape the process of grounding: the purpose, that is what people are trying to accomplish in the communication, and the medium of communication, that is the techniques available in the medium for accomplishing that purpose, and what it costs to use them. As Grice acknowledged [69], people in conversation generally try to establish a collective purpose. If the addressees want to understand what his or her partner wants to say to *a criterion sufficient for current purpose*, then the criterion must shift as their collective purpose change and, as a consequence, also the technique they use change basing on different content. There are two main types of contents analysed in Clark and Brennan's study [38]: reference and verbatim content. Conversations that focused on objects and their identities are very common in everyday life. The purpose of these conversations is to establish referential identity that is *the mutual belief that the addressees have correctly identified a referent.* There are many techniques for establishing this, for example the *alternative descriptions* used when the listener wants to demonstrate the partner that he or she have identified the reference object. A second technique are *indicative gestures* used by the partner for giving positive evidence that they have identified it by pointing. A fourth technique are *referential instalments* which is important to make clear the object that the speaker is about to present to avoid the incomprehension of the utterance. As a consequence, the speaker *can secure the reference by treating it as an instalment of the utterance to be confirmed separately* [38]. In English, there is a specialised construction for just this purpose called *left-dislocation*, as in *Your dog he just bit me.* This example, reported by the authors, begins with a left dislocated noun phrase, *your dog*, followed by a full sentence with a pronoun, *he*, referring to the same object. Another type of technique are the *trial references* that is a question in the middle of the utterance used to receive a positive or negative feedback from the interlocutor about the correctness of the presented information. There are some specialised situations where it is important to register the verbatim content of what is said, for example, when a speaker gives their telephone number, the address, and so on. New techniques have evolved for these kinds of conversation. The first are *verbatim displays* used by the partner to confirm what the speaker has just said, for example, by repeating a phone number. The second technique is through the *instalments* used when there are too many information to register verbatim and, as a consequence, the speaker cuts the utterance into instalments and receives verbatim displays on each instalment. The third is *spelling* the critical words that the speaker says for getting the verbatim content right. To summarise, according to the different purpose or medium, people use different kind of techniques in order to achieve the perfect understanding and to ground the new information correctly. In other words, the grounding process changes with the current purpose of that kind of conversation [38]. What is worth to highlight is that the effort employed in communicating changes is based on the communication medium. One type of technique available in a certain communication medium may be not available in another one, or even it may cost more effort in one medium rather than another. As stated by Clark and Brennan [38, p. 14], *people should ground with those techniques available in a medium that lead to the least collaborative effort.*

Many constraints are imposed by different media of communication between speakers, and this affects grounding as well. Clark and Brennan, in their study, present eight different constraints:

- Co-presence. The speakers share the same physical environment; hence they can look at each other and see what the other is doing.

- Visibility. The speakers are visible to each other, but they cannot see what the other is doing or looking at, for example through video teleconferencing.

- Audibility. The speakers communicate by speaking, they can hear each other, but they cannot take note of timing or intonation.

- Co-temporality. In face to face conversation, an utterance can be produced in the same moment in which it is received and understood. In other media, such as letters or emails, it is not possible.

- Simultaneity. The speakers can send and receive the message simultaneously, for instance when one speaker smiles while the other is still speaking.

- Sequentiality. Speakers' turns cannot get out of sequence in face to face conversation, while in emails or letters could be easy to find.
- Reviewability. A speaker cannot review the partner's message in everyday conversation because it tends to fade rapidly, but through other media such as letters or emails the message remain on the screen and can be review by both speakers or even by a third party.
- Revisability. A speaker can revise messages for the contributor. Some media, such as letters and email, allow a participant to revise an utterance privately before sending it to a partner. In face-to-face and telephone conversations, most self-repairs must be done publicly.

Grounding techniques, thus, can change according to the type of medium. The media differ in the cost they impose on some actions. Clark and Brennan [38] distinguished formulation cost and production cost for the speaker, and reception cost and understanding cost for the addressee. The formulation cost varies according to the type of utterance the speaker wants to produce: it costs more to formulate a complicated and perfect utterance, use uncommon words, and refer to unfamiliar objects. Production cost, conversely, varies according to the type of medium the speaker is about to use: using a computer keyboard or typewriter requires more effort than communicating face-to-face through eye contact and gestures. As far as the problem of grounding in human-machine interaction is concerned, the evidence for grounding can be very difficult and can require a major effort by the user. This is what Brennan [23] calls *the grounding problem in human-computer interaction.* Many of the problems that arise in human-computer conversation are due to inadequate feedback and impoverished context. As in human-human conversation, also with computers the timing of feedback is fundamental so that the conversation could be successful. Because of the obvious asymmetries in the capabilities of human and computer partners, most of the responsibility for coordinating joint activities with systems and for minimising the effort falls on users. Furthermore, dialogue systems tend to signal mainly negative evidences of understanding and minimal positive evidence; they also act as if most of their responses will be acceptable to users by not seeking evidence of acceptance from users and not providing any way to initiate clarification sub-dialogues. If language-based interfaces are to support mixed initiative dialogues (in which either user or system can flexibly take the initiative), then they need to support the systematic exchange of both positive and negative evidence [23]. On the other hand, reception costs for the addressee can be harder if the message is read rather than heard. In case of abstract arguments or complicated instructions, on the other hand, it may be perceived as easier to read rather than to listen. The context is, therefore, always important to determine such costs. Moreover, the understanding cost could be due to missing contextual clauses. For example, emails are not temporal nor sequential, *that makes understanding harder because the addressee has to remember what the message is in response to, even when the "subject" field of the message is filled in* [38, p. 17]. The other costs that the authors suggest in their study are paid by both speakers. The first the authors analyse are the *delay costs*, which are higher in face-to-face conversation, due to the fact that if the speaker takes too long before starting a turn or makes a pause too long in the middle of it, they can be misheard as dropping the conversation or as having finished the turn. The second is the *asynchrony costs*: people can time their utterance precisely in face-to-face conversation, but, through media without co-presence, timing is much less precise, and, without co-temporality, it is altogether impossible. Thus, grounding techniques that rely on precision of timing increase in cost when production and reception are asynchronous. Other costs are constituted by *speaker changes*, that are low in a face-to-face conversation where speakers follow the rule of one speaker at a time, but costs become higher, for example, in keyboard teleconferencing, where the points of speaker change are not as easily marked or readily recognised. The *display costs* are paid by both speakers as well. In face-to-face conversation is easy to monitor the facial expression of the interlocutor and grasp the moment of our turn. On the contrary, in media without co-presence, gestures cost a lot and are severely limited. There are costs associated with producing an utterance fault, that is, any mistakes or missings. Moreover, the cost of most *faults* depends on what it costs to repair them or to prevent them in the first place. In conversation, the production of speech is so spontaneous that a speaker may expect a fault from the other speaker. In email, faults are not as easily justified, because the sender has already had a chance to revise them, and also because the damage done is not as easily repaired. *Repair costs*, finally, tend to be minimal in audible conversation, where the speaker can repair the utterance as soon as they detect the fault; instead, in media that are not co-temporal, repairs initiated or made by others become very costly. In this case, speakers will try hard to avoid relying on others to repair misunderstandings. It is less costly for them to revise what they say before sending it. Another way to minimise repair costs may be to select a different and more appropriate medium. People, as we have seen, communicate through different media, but they do not do so in the same way. There are different costs that have to be paid by the speakers according to different media of communication. Clark and Brennan [38, p. 19] acknowledge that

> *"In media that are not co-temporal, there is an additional problem that A does not have immediate evidence as to which of these states B is in with respect to A's utterance. In a medium such as email, B's lack of response can be highly ambiguous. Did she not get the message, did she get it and not read it, did she read it and choose not to respond, did she not understand it, or what? A does not know whether B is in state O, 1, 2, or 3."*

That means media have different profiles of grounding costs: for example, over the telephone, it does not cost much to produce an utterance or change speakers. On a keyboard, instead, it costs much more. In the same way, repairs on a keyboard have a higher cost and for this reason the speaker tries to formulate their utterances more carefully, increasing their collaborative effort. Thus, it can be argued that speakers choose the medium according to their purpose of communication. Which medium is the best for which purpose then depends on the form grounding takes in a medium and whether that serves the participants' purposes.

What it is interesting to further investigate is therefore to study how grounding works in human-machine interaction when understanding problems occurs, specifically as far as inconsistency-driven misunderstandings are concerned. In this case, the accumulation of information in the common ground can head to the clashing items, where the presupposed one is the proponent, whereas the new evidence is the opponent. These clashing items constitute what argumentation is about, that is the *dialogue between two dialectical parties or roles* [97]. This dialectical approach will be described in detail and presented as a framework proposal in Chapter 3.

In conclusion, it is fundamental to underline again the importance of the grounding process: once we have formulated an utterance, we do not just utter it, but we have to make sure that the message has been perceived as we want it to be perceived. Nevertheless, misunderstanding and incomprehension can always occur and sometimes asking for clarification is fundamental in order to ground the new information correctly.

2.2 Clarification Requests as Grounding Tools

Clarification requests (CRs) are one of the pragmatic tools used in conversation to prove, ensure, and maintain the mutual understanding of the communicated message between the interlocutors. Purver [122] stated that interlocutors initiate a CR when a problem in processing the previous utterance occurs. For this reason, they are also called *anaphoric feedback* – they refer to what has previously been uttered. Furthermore, CRs are meta-communicative tools since they function as an acknowledgement of the level of understanding of an utterance [65]. The use of CRs is also described in terms of cognitive-pragmatic instruments to *ground*. As pointed out by scholars such as Clark [37], to pursue the aim of succeeding in their joint activity, interlocutors need to *ground* what is being communicated. In conversation analysis, *grounding* refers to the act of establishing that what we intend to say (or what has been said) can be well understood (or has been well understood) [38]. In order to do that, different strategies, such as linguistic or para-linguistic feedback analysis [163], can be exploited, among which we also find CRs. Among the scholars who pursued the intent of classifying different types of CRs, we mention Purver [122], who classified CRs according to form and reading (reported in details in Section 2.2.1), where *form* refers to the surface form, such as when an element from the previous utterance is used in the request (*reprise*), in combination with a 'wh-' interrogative pronoun (*wh-reprise*), or when a reformulation or a generic question is adopted (*non-reprise*). *Reading*, in contrast, refers to the compromised item which request questions about, such as a constituent or a clause. This classification established a precise way to describe how CRs can be automatically recognised or selected by a system and opened the way to further investigations, also concerning the causes and problems triggering the initiation of such requests. In [127], for instance, the notion of *problem*, causing the instantiation of a CR, was introduced. In fact, different kinds of problems, such as acoustic or lexical ones can determine the adoption of a different informative CR.

2.2.1 Related work

Clarification is a fundamental part of the grounding process. Through the pragmatic tool of clarification requests (CRs) the interlocutors can maintain the mutual understanding of the communicated message during a conversation. A clarification request is asked by a speaker only when they did not (fully) understand or are uncertain about what the previous speaker said or meant with an utterance [61]. Clarifications are uttered in a context of miscommunication. Following [74], miscommunication can be partitioned into three different types: Misunderstanding, non-understanding, and misconception. Misunderstandings are not detected immediately because the hearer thinks that what he or she has understood is the right message, but it is not the one the speaker intended to convey. The second type of miscommunication is non-understanding that occurs when the hearer finds the message uttered by the speaker ambiguous, that means with more than one interpretation, or, as Gabsdil [61] noticed, when they are uncertain about the interpretation that they gave to that message. In this case, even the form in which the requests are formulated can vary. Uncertain interpretations can coarsely be associated with single polar questions, whereas ambiguous understanding is more likely to result in alternative questions or wh-questions. Furthermore, non-understanding in general can occur on several different communicative levels, ranging from establishing contact among the dialogue partners to the intended meaning or function of the utterance in context. Clark [37] argues for four basic levels of communication in a framework that views the interaction as a joint activity of the dialogue participants. Clark's four levels are execution/attendance, presentation/identication, signal/recognition, and proposal/consideration. As Gabsdil [61] points out, on the lowest level, dialogue participants establish a communication channel, which is then used to present and identify signals on level two. On level three, these signals are interpreted before their communicative function is evaluated on the proposal/consideration level. The framework of joint actions requires that dialogue participants coordinate their individual actions on all of those levels. Gabsdil [61] in his study combines the cause of non-understanding with the Clark's four levels of communication, giving some examples and making a sort of coarse-grained classification of clarifications.

These examples of clarification requests correspond to two main readings for clarifications proposed by Ginzburg and Cooper [64]. Their "clausal reading" can be related to the presentation/identication level and their "constituent reading" to the signal/recognition level. Clausal readings *is commonly used simply to confirm the content of a particular subutterance* [64], it can therefore roughly be paraphrased as "Are you asking/asserting that X?" or "For which X are you asking/asserting that Y?". Constituent readings, on the other hand, *elicit an alternative description or ostension to the content (referent or predicate etc) intended by the original speaker of the reprised subutterance.* [64]. Misconceptions, finally, occur when the *hearer's most likely interpretation suggests that beliefs about the world are unexpectedly out of alignment with the speaker's* [74]. Clarications in response to misconceptions usually convey extra-linguistic information like surprise or astonishment. As we have seen, CRs can occur in different forms and readings, what they have in common is their property to be utterance-anaphoric, they concern the content or form of a previous utterance that has failed to be fully comprehended by the initiator [123].

Purver [122], identifies the various forms in which CRs can occur, illustrating them with examples taken from the BNC corpus, which contains a 10-million-word sub-corpora of English dialogue transcriptions about topics of general interest. Purver analysed a portion consisting of ca. 10,600 turns, ca. 150,000 words. The major CRs found in the corpus are reported below.

Non-reprise Clarifications The CR initiator does not make reprise of the information that has been misunderstood. On the contrary, the request is spelt out explicitly for the addressee and comes in the form such as *Do you mean... ?*, *Did you say... ?*.

Reprise sentences The speaker formulates a CR by repeating the previous utterance in full (A: *I spoke to him on Wednesday, I phoned him.* B: *You phoned him?*). The reprise can be also non-verbatim due to the presence of phenomena such as VP ellipsis or anaphora (A: *Oh, he's started this other job.* B: *Oh, he's started it?*).

WH-Substituted Reprise Sentences CR in this form is uttered when the speaker repeats the sentence in full but substitutes the unclear element with a wh-phrase (A: *He's anal retentive, that's what it is.* B: *He's what?*). Even in this case, the repetition may not be verbatim, but with the presence of an anaphoric or elliptical element.

Reprise Sluices This form is characterised by a bare wh-phrase used to reprise a particular phrase in the source utterance (A: *eon, Leon, sorry she's taken.* B: *Who?*). can be considered as a continuum of forms between wh-substituted reprise sentences and reprise sluices.

Reprise Fragments Elliptical bare fragment is used to reprise a particular phrase in the source utterance (A: *There's only two people in the class.* B: *Two people?*)

Reprise Gaps The difference between Reprise Gaps and Reprise Fragment is that the second does not reprise the element that needs to be clarified, but the immediately preceding component (A: *Can I have some toast please?* B: *Some?*)

Gap Filler This form can be useful when the speaker wants to suggest material which might fill a gap left by a previous incomplete utterance (A: *I'm pretty sure that the...* B: *Programmed visits?*)

Conventional A conventional form indicates a complete breakdown in communication. It is characterised by forms such as *What?*, *Eh?*, *Sorry?*.

Purver [123], after classifying the major forms encountered in the analysed corpus, shows the readings that CRs may present, following the study of Ginzburg and Cooper [64] we have mentioned in the previous paragraph. The main readings proposed by the authors are clausal/constituent/lexical, with the addition of the reading for corrections. Clausal reading, as Purver points out, *[...] takes as the basis for its content the content of the conversational move made by the utterance being clarified* [123, p. 239]; for the constituent reading, instead, it is meant the content of a constituent in the previous utterance that need to be clarified; the lexical reading is different from the clausal reading in the surface form of the utterance that need to be clarified rather than in the content of conversational move; the last reading, the correction, is paraphrasable as "Did you intend to/should you have uttered X (instead of Y)?". This is, therefore, similar to lexical reading in that *it queries surface form rather than semantic content but is distinguished by the fact that it queries a possible replacement or substitution of one part of the original form with another* [123, p. 239]. In view of these classifications, the author has analysed the correlation between form and function and, thus, he has observed the following results. The non- reprise form, for example, appears to be able to carry any readings, but the lexical one seems to be the most uncommon. Regarding the other forms, seems that many readings can be available for them. The most significant correlations are those of literal reprises that take only clausal readings, fillers, on the contrary, only lexical readings, wh-forms only the clausal or lexical, fragments and sluices usually take the clausal and, finally, conventional CRs usually have lexical readings.

The correlation between form and function of CRs has been investigated in a deeper way by Rodriguez and Schlangen [127], presenting a multi-dimensional classification of CRs forms and a fine-grained correlation between them and their functions. The study has been carried out in a corpus of German task-oriented dialogues, the Bielefeld Corpus, which contains 22 dialogues consisting of about 3962 turns, and 36,000 words. In the experimental setup, a dialogue participant was supposed to give instructions to the interlocutor to build a model plane. The authors pointed out some features used to describe the surface form of CRs. The first are the possible values of the attribute *Mood* which are declarative, polar questions, alternative questions, wh-questions, imperative and other; the values for the attribute *Completeness* that are particle (*Pardon?*), partial (a syntactic fragment) or complete (a syntactically 'complete' sentence); moreover, the possible values for the attribute *Relation* to the antecedent are literal repetition of the unclear part, the addition of a part to the repetition, reformulation of the problematic utterance or independent, that means no part of the utterance are repeated or reformulated. At last, the values for the *Boundary tone* are characterised by rising and falling intonation. Rodriguez and Schlangen posed the basis for the identification of problems that could cause misunderstanding, taking into account the CRs readings proposed earlier, but trying to define them in a more fine-grained way. The authors devised a multidimensional classication scheme where form and function are meta-features taking sub-features as attributes. They start from the models of Clark [37] and Allwood [3] about the four levels of communication mentioned before, adding other types of sub-levels. Each of those levels is a possible locus for communication problems. This dimension specifies the extent and severity of the problem. The extent, as Rodriguez and Schlangen argue, describes whether the CR points out a problematic element in the problem utterance or not. The severity, on the other hand, describes which action the CR initiator requests from the interlocutor: the CR initiator can ask a reformulation of the problematic utterance, probably triggered by a complete understanding failure, or they can ask a confirmation of the previous hypothesis of which they are not certain. The scholars classified also the reply to CR that can be a yes/no answer, a repetition or reformulation of the unclear element, an elaboration of the problematic utterance with the addition of new elements, a word definition, that is an answer to a lexical question (*what does x mean?*), or, lastly, no reaction at all. As a consequence, the satisfaction of CR initiators to the reaction of the CR addressee can be happy or unhappy, according to the right or wrong interpretation of the CR.

For the purpose of this study, it is fundamental to observe how CR behaviour in task-oriented dialogue differs from that in everyday conversation. Starting from the form and function classifications of CRs established in previous studies [121, 127], Rieser et al. [124] analysed the naturally occurring CRs in task-oriented dialogue, the human-human travel reservation dialogues available as part of the Carnegie Mellon Communicator Corpus, comparing their results with those of Purver [121] and Rodriguez Schlangen [127]. The first outcome is from the comparison between Communicator and BNC corpora. The authors assumed the hypothesis that different dialogue types lead the speakers to use a different grounding strategy. Grounding in task-oriented dialogues seems to be more cautious than in everyday dialogue. This hypothesis in supported by the fallowing facts:

- CRs are more frequent in task-oriented dialogues;
- The overwhelming majority of CRs directly follow the problematic utterance;
- Rs in everyday conversation fail to elicit a response nearly three times as often;
- Even though dialogue participants seem to have strong hypotheses, they frequently confirm them;

- Most CRs are partial in form;
- Most of the CRs point out one specic element;
- In task-oriented dialogues, the CR-initiator asks to confirm an hypothesis about what he understood rather than asking the other dialogue participant to repeat her utterance;
- he addressee prefers to give a short y/n answer in most cases.

The second outcome has been observed in the comparison between the task-oriented dialogues in Bielefeld and Communicator corpora. In there, several differences were found, even if less significant than the previous. These results showed that the process of grounding considers the dialogue types, modality, and channel quality. The authors noticed that the modality of the dialogue could change the source of understanding problem. For example, in corpora with a shared point of view (Bielefeld corpus), deictic reference resolution seems to create the major misunderstanding between speakers, whereas, on the contrary, in telephone communications (Communicator corpus), just one instance of this type was detected. Moreover, acoustic quality can create acoustic problems that appear to be more frequent in Communicate corpus, rather than in Bielefeld one.

2.2.2 Classification

Based on these studies and on an analysis of the SaGA corpus [100] (Chapter 3), a more fine-grained hierarchical classification is proposed in this work, as the first original contribution of this thesis. Starting from Allwood and colleagues' [4] four basic communicative functions, the communication levels *contact*, *perception*, *understanding*, *intention* were defined. At each level, one or many problems can occur, which are triggered by specific linguistic or informational issues. CRs can occur in different forms, i.e., wh-questions (WQ), alternative questions (AQ), positive polar questions (BroadPQ, NarrowPQ), negative polar questions (low or high [56]), declarative sentences (with or without a positive or negative question tag), or imperatives. Each formulation can convey a specific function and refer to a compromised item in the previous utterance. The order of the levels and of the triggers must be interpreted hierarchically: Contact ← Perception ← Understanding ← Intention. For instance, when a problem at the contact level occurs, all the other levels fail, as they are entailed in the first one; when a problem does not occur at contact level, it can occur at the perception level, and the following ones are, therefore, failing too, and so on. The classification illustrated in this chapter is applied to corpus analysis, as shown in Chapter 3, where the use of CRs is examined in an argumentation-based perspective, as when facing with conflicts in interaction.

2.2.2.1 Contact

With contact, we refer to the physical channel and psychological stance of the interlocutors to stay in communication. When the contact is not established, clarification requests can be used as a corrective feedback to restore the damage caused by the contact interruption. Lack of attention is the problem causing this interruption.

Communication Level	Problem	Trigger	Form	Function	Compromised Item
Contact	Lack of attention		WQ Conventionalised_forms	Repetition	Channel

Table 2.1: CR classification at the Contact level

a) Problem

   Lack of attention: inattention, boredom, unwillingness to follow the speaker can cause the channel shutting and the consequent loss of the input reception.

b) Form

   WQ: wh-questions are normally used to ask for repetition, such as *what did you say?*

   Conventionalised_forms: in this category, several types of question can fall, such as *Eh?*, *What?* and *Pardon?*, which signal a complete breakdown in communication [122].

c) Function

   Repetition: clarification questions are used in order to ask for repetition.

d) Compromised item

   Channel: what is compromised is the physical transmission medium and/or the psychological stance of the hearer.

2.2.2.2 Perception

Once the contact between the interlocutors has successfully been established, it could happen that the signal is disturbed for whichever reason. In order to signal this problem and to recover the missed information, clarification requests can be used by the hearer. Acoustic problems are the cause of such communication failures.

Communication Level	Problem	Trigger	Form	Function	Compromised Item
Perception	Acoustic		WQ BroadPQ NarrowPQ TagPQ AQ Declaratives Imperatives Conventionalised_forms	Repetition Confirmation	Signal

Table 2.2: CR classification at the Perception level

2.2.2.3 Understanding

Once the message has been well perceived, there could be problems of understanding, as far as the meaning, reference, syntactic and logical structure, or the completeness and coherence of the information are concerned. The listed triggers are intended to occur hierarchically, as previously mentioned.

Communication Level	Problem	Trigger	Form	Function	Compromised Item
Understanding	Lexical Understanding	Unknown_Meaning (MEA) Meaning_Ambiguity (AM)	WQ AQ BroadPQ NarrowPQ TagQ LowNPQ HighNPQ Declarative (Imperative)	Clarification Metalinguistic_function Disambiguation Confirmation Interactional	Clause Constituent External Presupposition
	Reference Reconstruction	NP_Reference (NP_Ref) Deictic_Reference (De_Ref) Action_Reference (Act_Ref) Elliptical_Ambiguity (ElA)			
	Syntactic Understanding	Analytical_Ambiguity (AnA) Attachment_Ambiguity (AtA) Coordination_Ambiguity (CoA) Elliptical_Ambiguity (ElA)			
	Logical Understanding	Cause_Effect (LOG)			
	Information Processing	Missing_Information (Miss_Inf) **Common Ground (CoG)**			

Table 2.3: CR classification at the Understanding level

a) Problem

Lexical Understanding: problems occur at this level when some lexical items are unclear to the hearer, who therefore asks for clarification.

- Triggers

  Meaning (MEA): the meaning of a lexical unit is unknown to the interlocutor, as in *Spun yarn? What does it mean?*.

  Ambiguity (AM): the meaning of a lexical unit can be ambiguous in a specific context (ex. A: *She is looking for a match?*, B: *A match?*)

Reference Reconstruction: uncertainties in the resolution of anaphora or extralinguistic reference can lead to clarification needs [127]. The reconstruction may concern a noun phrase, a deictic element, or an action.

- Triggers

NP_Reference (NP_Ref): this reference reconstruction problem is triggered by noun phrases, which may ambiguously refer to different linguistic or extra-linguistic items, i.e., A: *Please give me the double torx.* B: *Which one?*

Deictic_Reference (De_Ref): the problem is triggered by deictic expressions (personal, temporal, locative) which might refer to different linguistic or extra-linguistic elements, i.e., A: *Sarah brought a book for her sister. It's her favourite book.* B: *Whose?*

Action_Reference (Act_Ref): some actions can trigger understanding problems since their interpretation depends on their ambiguous relation with other constituents in a sentence (ex.: A: *Every wire has to be connected to a power source.* B: *Each to a different one, or can it be the same for every wire?*) [140].

Syntactic Understanding: understanding problems can be caused by a problematic recognition of word or phrases boundaries (i.e., *I saw a man with a telescope*) and syntactic structures. These problems can be triggered by different types of ambiguities.

- Triggers

Analytical Ambiguity (AnA): it occurs when the role of the constituent in a phrase or a sentence, such as the type of modifier in a noun phrase, is ambiguous [71], i.e., A: *the Tibetan history teacher can be introduced*, B: *Does he teach Tibetan history, or does he come from Tibet?*.

Attachment Ambiguity (AtA): it occurs when a particular syntactic constituent can be attached to two different parts of a sentence without compromising its grammaticality; the ambiguous constituent can be a relative clause, or, most commonly, a prepositional phrase, which can modify both a verb and a noun, as in A: *The police shot rioters with guns*, B: *Who had guns? The police or rioters?* [71].

Coordination Ambiguity (CoA): it occurs when i) more than one conjunction and or or is used in a sentence, as in A: *I saw Peter and Paul and Mary saw me* B: *Who saw you then?*) or ii) one conjunction is used with a modifier, as in A: *I saw young men and women at the demonstration*, B: *Young men and young women?* [71].

Elliptical Ambiguity (ElA): it occurs when it is not clear whether constituent is omitted or not resulting in syntactic ambiguities in the sentence reconstruction, as for *Perot knows a richer man than Trump* which can be interpreted as not elliptical (Perot knows a man who is richer than Trump is), or as elliptical (Perot knows a man who is richer than any other men that Trump knows) [71].

Logical Understanding: this problem refers to the logical relation connecting the new information to the antecedent one [140]. This can, indeed, not always be clear. This problematic relation is mostly causal.

- Triggers

  Cause_Effect (LOG): the cause-effect relation is not clear, as in A: *Max fell. John pushed him.* B: *Witness, do you mean that he fell because he was pushed by the defendant?* [140].

Information Processing: this problem refers to two different issues concerning the received information, as i) it could not be satisfactory for its entire understanding, or ii) the previously grounded information needs to be stabilised or checked because of inconsistencies via a confirmation or a control-targeted question.

- Triggers

  Missing Information (Miss_Inf): when the understanding of some input is based on something which is not yet part of our common ground, we can use an information-seeking question to recover the needed data to correctly process the previous utterance, as in A: *When you see the building on your left, you can proceed on the right.* B: *Which colour is the building?* In this case, the question is not biased, in that the speaker does not have a clue on the possible answer. Conversely, they believe that more information might be useful to better use the previous one. Missing Information CRs can be distinguished from other information-seeking questions, in that they always refer to some aspects of the previous utterance and therefore they do not cause a change in topic.

  Common Ground (CoG): when some information is eligible to be part of the common ground or when it is already part of it but clashes with current inputs, we may find ourselves to i) pose a question to check if the complex information (that is an information which may cause problems in the future dialogue states or whose modification could request some efforts) is well-understood before storing it in the common ground (confirmation request), or ii) pose a question to check if the previously stored information is correct, as it shows inconsistencies with respect to the current dialogue state (clarification requests). This category of problems is similar to what was described in Schlöder et al. [141], as far as the intention adoption is concerned. In fact, the authors describe this category as an impasse in the acceptation of the speaker meaning, because of an incompatible belief. We can here argue that, this incompatibility is here represented in the construction of the listener knowledge in the Personal Common Ground, rather than at an intentional level. Further examples and explanation will be provided.

b) Form

WQ: wh-questions are used to check the understanding of a previous utterance or part of it, in order to ask for explanation, or disambiguation, as in *What does it means?*

AQ: alternative questions are adopted when two or more items need to be disambiguated or when hypotheses are presented in a Missing Information scenario, as in *Do I need to go left or right?*. As a matter of fact, Rieser et al. [125] claim that hypotheses are preferentially used by speakers when asking for clarification.

BroadPQ: broad polar questions refer to questions used by speakers to ask for confirmation or clarification showing different degrees of bias toward the information asked, i.e., A: *I had my hair cut today*, B: *Were you not at school?*

NarrowPQ: narrow polar questions refer to elliptical questions used to ask for confirmation for understanding reasons, such as when the word is unknown A: *Do you want some Sauerkraut?*, B: *Sauerkraut?*

TagQ: tag questions consist of an *anchor* and a *tag*, where the anchor presents a hypothesis, such as the grounded information, and the tag is used for asking for confirmation, as in A: *Turn to the right to go to the restaurant*, B: *You said to the right, don't you?*

LowNPQ: low negation polar questions refer to questions which contain a sentential negation [88] used for confirmation purposes, as in *For the party, we need nine hamburgers*, B: *Is John not coming?*

HighNPQ: high negation polar questions refer to questions whose negation is used to check the positivity or negativity of a proposition, reflecting the speaker beliefs toward it [88], as in A: *For the party, we need ten hamburgers*, B: *Isn't John vegetarian?*

Declarative: declarative sentences can also be used, potentially with a rising intonation, to ask for confirmation.

c) Function

Clarification: requests triggered by understanding problems can be used to seek for clarification that is the addition of further information to the previous utterance [136]. In some cases, clarifications can become explanations when the gricean maxim of quantity is not applied, as the speaker feels the need to give more information than request for understanding purposes. In Schettino (2020), it was pointed out that clarification speech acts are the most frequently used ones after implicit and explicit CRs.

Metalinguistic_function: the meta-linguistic function refers to clarifications about the linguistic items which need to be clarified. It is, therefore, a sub-type of clarification concerning the linguistic code. For example, this function is the one occurring when word meanings are not clear.

Disambiguation: clarification of ambiguities can occur at the lexical, referential and syntactic level.

Confirmation: when the speaker has some kind of understanding hypothesis, a CR can be used to seek for its confirmation or denial [125, 62].

Interactional: the interactional function is characteristic of CRs which point out feedback about the speakers' stance, in that the mental state is represented. In other words, this function occurs when the speaker sheds a light on current presuppositions toward the grounded knowledge to be questioned about.

d) Compromised item

Clause: because of some unclear clause in the previous utterance, CRs can be used, as in in A: *I saw Peter and Paul and Mary saw me.* B: *Who saw you then?*

Constituent: the understanding problem regards a specific constituent in the previous utterance, as in A: *I want to leave from Potsdam.*, B: *From Potsdam? To where?*

External: an external compromised item is some information which is needed and not part of the previous utterance, as for Missing Information CRs.

Presupposition: presupposition refers to the propositional attitude of speakers towards previously grounded knowledge [155]; when it is compromised, the previous discourse context shows some inconsistencies with the common ground; for this reason, the speaker asks questions which might or not underline speaker beliefs towards it (see Bias).

2.2.2.4 Intention

Once the message has been received, acoustically, semantically, syntactically understood, and the information received is satisfactory and consistent with the common ground, other issues can occur as far as the real intention of the speaker is concerned. This can result in problems in the recognition or in the evaluation of the intention.

Communication Level	**Problem**	**Trigger**	**Form**	**Function**	**Compromised Item**
Intention	Intention Recognition	Inference	WQ AQ BroadPQ NarrowPQ TagPQ LowNPQ HighNPQ Declarative (Imperative)	Metacommunication Explanation Clarification Confirmation Disambiguation Correction Interactional	External Speaker_Meaning
	Intention Evaluation	Agreeing/Disagreeing Interest Incredulity			

Table 2.4: CR classification at the Intention level

a) Problem

Intention Recognition: the intention is not well recognised because of some inference problems; the speaker pose a question to verify the well processing of the inferential comprehension [140], as in A: *What time is it?*, B: *Do you want to leave?*

– Triggers

Inference: a misleading inferential processing, whose aim is to recover implicated intentions [154], can cause problems in intention recognition, as in A: *My mother is a lawyer*, B: *Are you trying to threaten me?* [140]; this problem results in a specific class of CRs called *speech act determination* [141].

Intention Evaluation: the recognised intention is commented to show one's own intentional response reflecting the personal stance, such as agreement, disagreement, interest, and surprise [140]. For example, in A: *You need a visa.*, B: *I do need one?* [141], depending on the intonation of B's dialogue act, the speaker might express surprise or disagreement about the speaker meaning of A.

- Triggers

  Agreement/Disagreement: the recognised intention can or cannot be adopted by the hearer because of compatible or incompatible beliefs at the intentional level, as for the category of intention adoption described in Schlöder [141], i.e., A: *That's a very unnatural motion.*, B: *Do you think?*

  Interest: this trigger does not really represent a critical issue, as it rather expresses the hearer's implicit interest to gather more information about the previous utterance; as a matter of fact, the hearer can be interested in something which represents a novelty resulting in a question expressing curiosity and surprise, and that implicitly bring the speaker to give further information concerning the topic, in form of clarifications or explanations.

  Incredulity: this trigger comments the recognised intention which is incompatible with the pre-existent beliefs; instead of expressing disagreement, the hearer just points out the incompatibility to what was thought.

b) Form

WQ: wh-questions are used to ask for clarification (meta-communicative or not) concerning the speaker meaning, as in *What do you mean?*

AQ: alternative questions occur when two or more speaker meaning hypotheses are presented to the interlocutor in order to verify the correctness of the inferential comprehension, as in *Is that a question or a request?*

BroadPQ: complete positive polar questions are used to confirm the inferential comprehension, as in A: *It's hot in here.*, B: *Are you saying you want me to open the window?*

NarrowPQ: elliptical positive polar questions reprise previous linguistic material to ask for clarification or explanation whenever an intention is not clear.

TagPQ: tag questions are adopted when a hypothesis needs to be confirmed.

LowNPQ: low negation polar questions are used to ask for confirmation.

HighNPQ: high negation polar questions are used to ask for different purposes, such as correcting or asking for clarification.

Declarative: declarative sentences can also be used, potentially with a rising intonation, to ask for confirmation.

c) Function

Metacommunication: the verbal activity expressed by the question is directed to the illocutionary dimension, that is to the pragmatic level; these questions, therefore, refer to what the speaker mean by what has been said [174], as in A: *Can you pass me the salt?* B: *Is that a question or a request?* [140].

Clarification: requests triggered by intention recognition problems can be used to seek for clarification that is the need for additional information to the previous utterance in order to proceed with the inferential processing.

Explanation: when clarifications need more information or when CRs are used to express interest or incredulity, the consequent function may be the need to get explanations rather than simple clarifications.

Confirmation: when one or more hypothetical inferences are presented, CRs express a confirmation function.

Disambiguation: when an utterance can have more intention interpretations, CRs are used to disambiguate.

Correction: this function refers to the possibility of a hearer to infer something which clashes with what has been uttered in a sense that the context enables the hearer to process the inference, although the speaker made a mistake, as in *Did you intend to refer to X (instead of Y)?* [122].

Interactional: this function refers to the possibility fulfilling important tasks for the discourse processing activities of the interlocutors, such as back-channel, turn-taking, pause-filling, attentive singals. CRs with this function are mostly used to express interest or incredulity.

d) Compromised item

Speaker Meaning: differently from the meaning, which refers to the semantic content of words, clauses and sentences, speaker meaning refers to the meaning intended by the speaker [10]; when the hearer is not sure or did not understand this meaning, CRs can be used to reconstruct this communicative item, as in *What do you want to say with that?*

External: an external compromised item is some information which is needed and not part of the previous utterance, as it needs clarification, explanation or else.

2.2.3 Common Ground Clarification Requests

The last sub-category of Understanding CRs – *Information Processing* – is particularly interesting when considering dialogues where one interlocutor gives information to another, who knows less about the topic, e.g., as in a map-task situation. In human–machine interaction scenarios, we can find comparable situations in what we call *User Guided Tasks*, i.e., those tasks where the user has a leading and the machine a following role. In such situations, the information given by the users are new to the receiving system. Although not having knowledge of the desired final state and of the steps to reach it, the system does have a general knowledge of which action is possible or not possible, in terms of pre-conditions and post-conditions (see Chapter 6). This is what belongs to the CCG [36], that is, the *rule-based* shared knowledge between individuals belonging to the same community. The *fact-based* knowledge built between two interlocutors, which is tied to the structures comprised in the CCG, is conversely what has been called PCG. Concerning the PCG, the system might need to complete some received information to store it in the PCG based on the rules of the CCG by asking specific clarification questions, which we call *Missing Information* CRs. Commercial and academic systems are already treating this system necessity, for example, through *slot-filling* strategies. On the other hand, when the information needs a double check before the storing in the PCG can take place (based on the rules of the CCG), or when the received information clashes with what we have already stored in the PCG, the system might need to use a *Common Ground* CR.

In Figure 2.1, the scenario eliciting a Common Ground CR is displayed. In the mind of the female agent A, the CCG is stored to guide the process of accumulating information in the PCG. The information (i_1, i_2, i_3, ..., i_n) are communicated by the male agent B to A, and sequentially stored in her PCG. When B utters a new information i_z, this is represented as a new item candidate to be part of the PCG. This representation has *disastrous* results, in that the presence of the new item i_z in the PCG clashes with the presence of another item i_3, whose validity is now questioned. This conflict represents a Common Ground Inconsistency and is translated in the Common Ground CR $\neg i_3$?, whose form, function and illocutive effect are analysed in the next chapters. As a preview of Chapter 3 and as already highlighted in the classification of CRs, it can be pointed out how important polar questions are to Common Ground Inconsistencies, in that their epistemic stance (or presup-

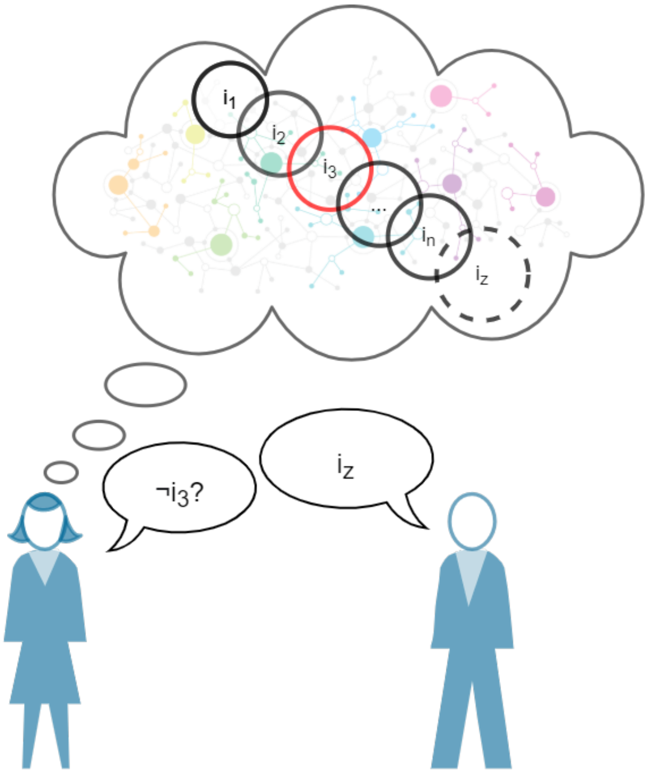


Figure 2.1: Representation of the Common Ground CRs elicitation scenario

positional stance) is clearly expressed compared to other types of questions. Finally, differently from other CRs, Common Ground CRs do not necessary refer to the immediately previous utterance, but to previously - correctly or wrongly - grounded information.

2.3 From Questions to Polar Questions

Questions are utterances that seek for verbals or other semiotic responses (i.e., a nod) [72, 149, p. 395]. In fact, questions have different interactional consequences: they bring interlocutors to elaborate an answer, impose presuppositions, agendas, and preferences, and cause various actions that might be potentially face-threatening [26]. Studies on questions demonstrate that there are three different aspects that are to be studied when dealing with interrogatives, that are grammar, prosody, and epistemic asymmetry, that is the different degree of expertise of the interlocutors toward a specific topic [148]. According to the type of question - content (wh-questions), polar, or alternative - these levels of analysis interconnect with each other to express specific functions.

In this work, grammar, prosody, and epistemic asymmetry of polar questions are described, as this type of questions appear to be mostly preferred when Common Ground Inconsistencies occur (Chapter 3). In general, polar questions can be defined as questions that make *relevant affirmation/confirmation or disconfirmation* [159]. Polar questions can have two possible binary answers: true versus false. Many languages have grammatical markings which distinguish polar questions from declarative sentences (word order, question particles, etc.). Each language, moreover, can also have many ways to ask polar questions [72, 149, p. 396]. In English, for example, we can generally have, as previously mentioned:

i positive polar questions (i.e., *Did he bring food?*)

ii high negation polar questions (i.e., *Didn't he bring food?*)

iii low negation polar questions (i.e., *Did he bring no food?*)

iv tag questions (i.e., *He brought food, didn't he?*)

Nevertheless, it is important to remember that these standard grammatical types are not specific for questions in any context, since an utterance in one of these interrogative forms does not necessarily do questioning, and on the other hand a non-interrogative utterance can also function as a question [72, 149, p. 396].

Interrogative prosody is another linguistic criterion which is used in most languages. In Italian, Arabic, and Romanian, for example, rising intonation is described as a conventionalised mean to ask polar questions [57]. More in detail, Italian is considered one of the 173 languages which have been mapped in *The Wolds Atlas of Language Structure Online* as a language with interrogative intonation as the only interrogative marker[1] [57]. However, it is misleading to consider rising intonation as a strongly indicative mark of polar questions. In fact, polar questions are not necessarily uttered with a rising intonation and a rising intonation is not necessarily used exclusively with questions [72, 149, p. 396]. Furthermore, there are languages that do not have specific grammatical or intonational resources to mark polar questions, as in a documented Papuan language, Yélî Dnye [95].

The way speakers interpret a question as a question depends, therefore, on other factors as well. An important criterion is indeed the domain of knowledge. For instance, Labov and Fanshel [90] stated that *when a speaker makes a statement about an event that falls into the recipient knowledge domain (B-event statement), it functions as a polar question and elicits confirmation or disconfirmation.* The authors made a distinction between *A-events*, which are known to *A* but not to *B*, and *B-events*, which are known to *B* but not to *A*. When *A* makes a statement which is not part of their domain of knowledge, the

[1] https://wals.info/feature/116A2/33.7/153.1 [last consultation on the 4th January 2021]

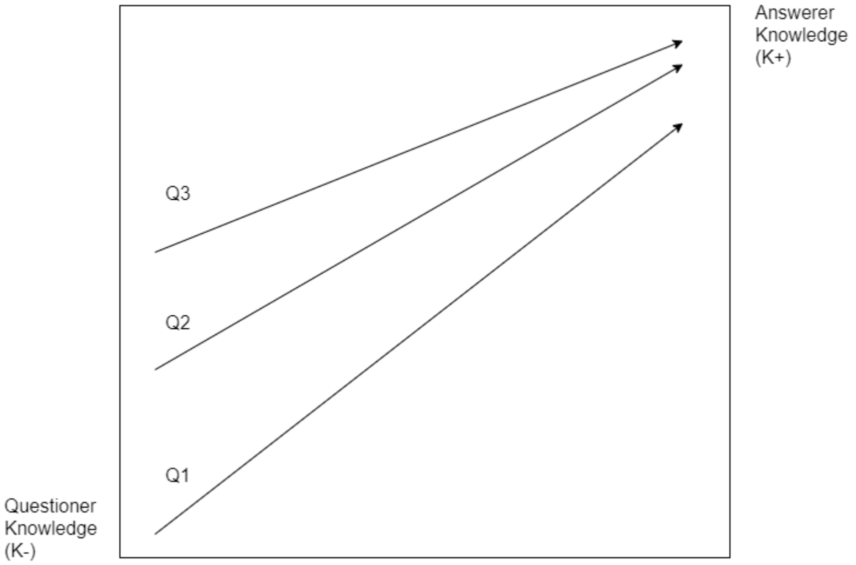


Figure 2.2: Question design and epistemic gradients (based on [72]). The epistemic stance of the questioner starts from a low lever of knowledge (K-), in the lower left corner, and slightly increases on the $y - axis$. Different questions correspond to different degrees of epistemic stance

statement will be interpreted as a question. Or in Levinson's words [95], *[. . .] when an utterance addresses information that the speaker does not know but a recipient is likely to know, it is treated and responded to as a polar question or a confirmation request* [72, 149, p. 397]. The *epistemic asymmetry* plays, therefore, a crucial role in the interpretation of questions. However, the epistemic stance of the questioner can correspond to diverse gradients which can correspond to different syntactic forms (Figure 2.2): Q1) *Who did you talk to?* - this content question suggests that the questioner has little knowledge about the topic (higher K-); Q2) *You talked to Steve?* - this declarative interrogative suggests that the speaker expects a positive answer, as they know more about the topic (lower K-); Q3) *You were talking to Steve, weren't you?* - this positive statement followed by a question tag suggests that the speaker strongly believes in their presupposition for which they just need a confirmation, that is a positive answer (higher K+) [72, 149, p. 399]. For this reason, the epistemic gradient and, therefore, the bias of the speaker towards a presupposition and a consequent expected answer also determines the grammar and prosody of the question. In fact, some studies also showed that falling intonation in polar questions is associated with higher certainty, whereas rising intonation with lower certainty [42].

2.3.1 Bias in Polar Questions

Since the epistemic stance can be encoded in the grammatical and prosodic form of polar questions, this means that these questions can function as an open door to the mental state of the questioner, who has a specific opinion about certain information. According to Oshima [114], polar questions convey an epistemic bias toward a positive or negative answer. In fact, in Bolinger's words [20], polar questions advance a *hypothesis for confirmation*, where the hypothesis can be positive (i.e., *I strongly believe that this is true*), neutral (i.e., *I don't know a lot about it, therefore I need confirmation on that*), or negative (i.e., *I strongly believ that this is false*). This means that questions do not only serve to request information, as they can also be employed as a powerful tool to control the answerer's reactions.

Polar questions can, therefore, convey three major constraints:

i Presuppositions: defined as the background beliefs of the speaker encoded in a statement whose validity is taken for granted [156], presuppositions are usually used in questions unproblematically; nevertheless, questioners can also embed hostile presuppositions in questions [72, 149, p. 401], in that they convey and impose on recipients questioners' beliefs.

ii Agendas: questions set agendas concerning the topic (what the questioner is talking about) and the action (what the questioner is doing with the question, i.e., suggesting an answer), which both can be biased [72, 149, p. 402].

iii Preferences: when questioners pose questions, they can set preferences, such as answers over non-answer responses, or affirmation over disaffirmation; concerning this last point, PQ forms typically display a preference for:

- affirmation[2]
  - positive polar questions: *Have you heard from her?*
  - positive statements combined with a question tag: *You've heard from her, haven't you?*
  - positive declarative questions: *You heard from her?*
  - negative polar questions: *Haven't you heard from her?*
- disaffirmation
  - negative declarative: *You haven't heard from her/You never heard from her*
  - negative statements combined with a question tag: *You haven't heard from her, have you?*
  - positive polar questions combined with a negative polarity item: *Have you heard from her yet?*

[2] In this work, the positive bias has been interpreted as the expectation of a positive answer depending on the positive epistemic stance towards a previous presupposition.

  - positive interrogatives combined with negatively tiled adverbs: *Have you really heard from her?*

[72, 149, p. 405].

As previously pointed out, when a speaker has a K- position, their utterance is recognised as a question. Besides this crucial factor, it can be noticed that a questioner has a position closer to the K+, in case they already possess that specific knowledge. In this work, this is the case of a previous grounded knowledge. When this happens, the question tends to be interpreted as a criticism or challenge [72, 149, p. 410]. As Steensig and Drew acknowledged [158], *asking a question is not an innocent thing to do; when a question is asked about what its recipient has said or done, it carries a possible implication of disaffiliation.* In the terms of this work, when a Common Ground Inconsistency occurs, the questioner, referring to the knowledge that is already stored in the common ground, challenges the answerer, who states something contradicting the questioner's K+. In this context, a Common Ground CR in the form of a polar question is uttered. This, in turn, expects a positive answer because of a strong belief towards that stored presupposition. In the next Chapters, further analysis on the use of such questions are carried out starting from a corpus analysis before experimenting their appropriateness and efficiency.

Chapter 3

A Computational Model of Common Ground Conflict Search and Signalling

> A man of the state of Chu had a spear and a shield for sale. He was loud in praises of his shield. "My shield is so strong that nothing can pierce it through." He also sang praises of his spear. "My spear is so strong that it can pierce through anything." "What would happen," he was asked, "if your spear is used to pierce your shield?" It is impossible for an impenetrable shield to coexist with a spear that finds nothing impenetrable.
>
> His Spear Against His Shield - 自相矛盾

The research on dialogue systems underlined, since the beginning, the need to test and evaluate their functionality and performances. Nevertheless, the evaluation of dialogue systems has always been a problematic task to carry out. When Turing [166] suggested the imitation game (Figure 3.1) as a possible evaluation of the intelligence a machine can show, he was thinking of replacing the question whether a machine is able to think with its imitation capabilities. In fact, the concept of thinking was thought to be difficult to define. Instead, the imitation game could actually be a valid and answerable question to pose. To answer this question positively, the evaluator should not be able to tell the difference between the machine and the human interlocutor, in that the machine succeeded in imitating intelligent human-like behaviour. Here, the concept of *intelligence* needs some in-depth consideration.

Gottfredson [68, p. 13] defined *intelligence* as the *ability to reason, plan, solve problems, think abstractly, comprehend complex ideas, learn quickly and learn from experience.* As we may easily comprehend, this definition is far from the possibility a machine can have to imitate some behaviours. If the aim is not only to reproduce, but also to evaluate some intelligent aspects a machine could have, we may need to adopt different tests. Therefore, the Turing test,

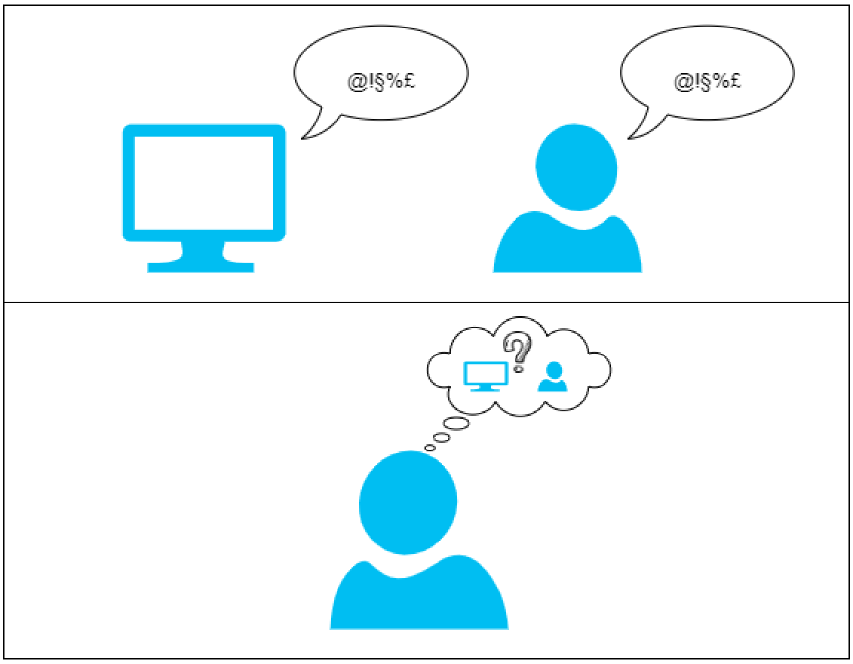


Figure 3.1: The imitation game consists of the ability of the machine to imitate the way human communicate; through this test the interlocutor is not capable of distinguishing whether he/she talking to a machine or to a human interlocutor

although sometimes still used, can conversely represent the desirable goal of an intelligent agent, which shows behaviours that are human-like, rather than an evaluation tool for system performances. At the same time, the question that could here be raised is whether we really want a system to be completely indistinguishable from human beings and why we want that. Conversely, we might want systems capable of showing their specific intelligent features which might be suitable for *artificial beings* only. Similarly to Turing, in [138], two aspects are evaluated: i) human-like system's responses; ii) how well the user models cover the variety of the user population in the training data. Even here, what is missing is a shareable framework to carry out this evaluation and an in-depth description of how the system is actually working.

Whichever is the way we imagine our dialogue system to be, the evaluation should rather consider some specific traits of what we call intelligence. For goal-oriented dialogue systems, the completeness of the task, dialogue length, and user satisfaction are usually taken into account. On the other hand, for general purpose dialogue systems, approaches like next utterance classification and word perplexity are preferred [144]. To the present day, fully satisfactory automatic classification metrics for dialogue systems does not exist. Nevertheless, the combination of different methodologies could lead to better results.

Starting from some human-computer interaction usability principles, we can point out some properties which can also be applied to the definition of what intelligence in machines might be. In [55], three main principles are listed as reported below:

1. Learnability: the ease with which new users can effectively interact with the system without encountering particular difficulties; it can refer to different aspects, such as
    (a) predictability, which is the use of affordance and logical constraints to indicate available actions;
    (b) synthesizability, which refers to the expected feedback or change in the state of a system occurring while the user is interacting with the system itself;
    (c) familiarity, which relies on the past experience with other systems;
    (d) generalizability, which refers to the possibility users have to extend their knowledge to situations that are similar but unknown.
    (e) consistency, which relates to similar behaviours occurring in alike situations or alike tasks.
2. Flexibility: the diversity of ways the user and the system can exchange information; diversity, for the purpose of this work, might also rely on the pragmatic adaptation to contextual needs, such as the choice between specific clarification requests; different parameters can be considered in the achievement of flexibility, as reported below
    (a) dialogue initiative, refers to the agent controlling the dialog flow, that is the human or the virtual one;
    (b) multi-threading, which is the ability to support simultaneous tasks (i.e. multimodality);
    (c) task migrability, which is the possibility to transfer the control of a task from the user to the system and vice versa;
    (d) substituitivity, which can be translated in the chance the user might have to change the execution of an action with different options;
    (e) customizability, which refers to the adaptation of the interface to the user's needs.
3. Robustness: it is the level of support that the system provides to the user in completing and assessing a task successfully; this can also be ensured by the ability a system can have to check message understanding and correcting alleged errors in order to successfully complete the required tasks; the following principles are applied to support system robustness

(a) observability, which refers to the possibility to observe the internal state of the system; this can be further represented by five different principles: i) browsability allows the user to explore the internal state without modifying it, ii) default suggests the user possible actions, iii) reachability enables the navigation through observable states, iv) persistence refers to the duration of an observable state, v) task performance includes the services supporting all the possible tasks;

(b) recoverability, which is the ability a system has to recover in case of errors; error recovery can act forward and backward: i) forward error recovery refers to errors in the current state causing a negotiation from that state to the desired one, ii) backward error recovery aims at correcting the effects of previous states in order to return to a preceding state; Common Ground CRs function, indeed, as backward inconsistencies recovery of grounded information;

(c) responsiveness, which is the time the system need to give feedback and communicate with the user;

(d) task conformance, which refers to the level of support a system offer in the execution of a task in an expected way.

Related to the robustness principle, it is with no doubt that systems can make their internal states observable through verbal or non-verbal interaction. Specifically, when problems occur in information processing, the observable character of such states can be utilised to recover the problems. However, to verbalise internal states, the system itself needs to have access to this information. For the accessibility of the information, whose consistency must be continuously checked, a knowledge representation is, therefore, needed. In Chapter 6, its structuring in a graph database is reported.

In this perspective, it can be theorised an experimental reading key of information processing machines dealing with the accessibility to their internal mental-like information states as a shareable intelligent trait [47]. This capability can be directly connected to the possibility that they can also have to express the presence of common ground inconsistencies, which is the topic of this work. This mental-like process brings us to the concept of *consciousness*, which is of *inspiration* for the modelling of some of the processes reported in this work, although without claiming to actually grasp and reproduce what consciousness really is. In fact, it is important to remember that consciousness has been defined differently according to the different scholars and disciplines dealing with it, as no shared or standard definition exists yet and many of them still lead to lots of controversies. At its simplest, it can be defined as the state of awareness of an internal/external condition or experience. In the field of artificial intelligence, many debates focused on the possibility for computational systems to show consciousness. To address this topic in a specific human-machine interaction application, we focus in this work not on the hard problem of consciousness, but on the weak one, represented by the concept of *Access Consciousness* (A-Consciousness). A-Consciousness is described as the conscious access to a mental state to reason about it for rational control of action and speech [18]. In other words, it represents the availability or accessibility of the content of a mental state for verbal reports. The mental states that are here accessible and, therefore, verbalised are the *propositional attitudes* (Chapter 1). A-Consciousness can be distinguished from *Phenomenal Consciousness* (P-Consciousness) which, conversely, is about the subject's perception of a conscious experience. Although Block's distinction is commonly accepted, it is important to mention that some other philosophers, such as Lycan [102], identified other possible fine-grained classifications of consciousness, such as the distinction between organism consciousness, control consciousness (similar to Block's A-Consciousness), consciousness of, state/event consciousness, reportability, introspective consciousness, subjective consciousness, and self-consciousness. The process involved in A-Consciousness occurs together with the Information Processing one. According to Block, *a perceptual state is access-conscious if its content is processed via that information processing function, that is, if its content gets to the Executive System, whereby it can be used to control reasoning and behavior* [18, p. 3]. This Executive System is what can be modelled in order to make some process available and, therefore, a source of reasoning. This information-processing centre is what in this study will be called *Conflict Search Graph*, representing one of the possible modules of an Executive System (Chapter 6).

In this work, we account for an information-processing system which applies processes inspired by the definition of A-Consciousness, in that it always has access to, or awareness of, its informational internal states and, moreover, produces information rather than just transmitting it, as it is capable of reasoning and acting upon its interpretation. It is firstly important to describe what information-processors are, as they represent one way to explain access-consciousness in both biological and virtual systems. Any system capable of taking information in one form and processing it into another is referred to as an information-processor. *Information* can be defined according to the processes that may be involved in the use of information itself. In [151], some of these processes are listed as follows:

- **external or internal actions triggered by information**,
- segmenting, clustering, labelling components within a structure (i.e., pars-

ing),

- trying to derive new information from the old (i.e., What caused this? What else is there? What might happen next? Can I benefit from this?),
- **storing information for future use (and possibly modifying it later)**,
- considering and comparing alternative plans, descriptions or explanations,
- **interpreting information as instructions and obeying them, e.g. carrying out a plan**,
- observing the above processes and deriving new information thereby (self-monitoring, self-evaluation, meta-management),
- communicating the information to others (or to oneself later),
- **checking the information for consistency**

Some of the aforementioned processes are here in bold, since they represent some crucial aspects of an example of information-processing systems, which we are interested in. Specifically, we can call them *User Guided Tasks Applications*. These applications require the user to have a leading role and the machine to have a following role. This type of task can be considered as a sub-type of *User-initiative tasks*. In addition to the characteristics typical of a User-initiative system, in our model, the system checks for consistency based on shared rules. In such situations, the information given by the users (building the PCG) is new to the receiving system, although the general knowledge of the domain are shared (CCG). The user has a higher K+ position with respect to the system, since the desired goal is known. Conversely, the system does not have the same K+ position. Nonetheless, the structured PCG is used to build presuppositions with strong confidence, that make the system closer to the K+ position to the point that, in case of inconsistencies, the system can assume the role of questioner (see Chapter 2 for further details on grounding and bias in polar questions). This means that such systems take information as input and store it for future use in a learning perspective, not only to carry out a plan in the future but also to build presuppositions for recollection and consistency checks. Such information can, therefore, be modified later, for instance, when inconsistencies between two pieces of information occur. In fact, specific corrective actions, such as clarification requests (see Chapter 2), can be triggered by inconsistencies in the common ground in order to disentangle them, therefore producing new information as in a conscious-like process [105]. Although a User Guided Task Application does not have knowledge of the desired final state and of the steps to reach it, the system: a) does have a general knowledge of which action is possible or not possible (i.e. CCG); b) can store the given steps in the contextual knowledge (i.e. PCG), where both knowledge structures are modelled in a graph database (Chapter 6). When inconsistencies arise because of unverified pre-conditions of current actions, and conflicts between previously produced post-conditions and current actions' pre-conditions, adequate linguistic actions can be adopted to solve the problem.

Not only does this pragmatic skill represent a process inspired by A-Consciousness, but it can also be a sign of how machines can simulate intelligent behaviour. Since scholars define intelligence in different ways, and different types of intelligence are thought to exist, we refer here to *interactional intelligence* (Chapter 1). This intelligence lets the system be argumentation-capable. The linguistic activity of argumentation is pragmatically regulated by a sequence of purposive speech acts in conflict [172]. In this Chapter, it is firstly referred to as an argumentation-based dialogue framework. Afterwards, it will be payed attention to linguistic strategies which can be adopted in one possible application of this framework. The validity of these strategies is retrieved from a corpus analysis, whose underlined linguistic forms will be further investigated in the next chapters.

3.1 Towards an Argumentation-based Dialogue Framework

In the Chinese story, *His Spear Against His Shield*, the man of the state Chu caused a passerby to pose a question because of his inconsistent advertising messages. The informational inconsistency was, therefore, cause of the beginning of a conversational exchange. In fact, a dialogue is originated from a conflict, as it is defined as "a discussion between two or more people or groups, especially one directed towards the exploration of a particular subject or resolution of a problem"[1]. Given that an *argument* is referred to as the problem of establishing the truth of a statement through the exchange of ideas, it can be deduced that argumentation is the essence of dialogues. The art of speaking as a method used to discuss opposing ideas to find the truth has been explored since ancient times. This art was called *dialectic*. It is heavily based on one-to-one exchange of ideas and it can be both interpreted in a competitive or in a collaborative way. This dual nature distinguishes dialectic from rhetoric and debate. On the one hand, *rhetoric* is defined as "the art of effective or persuasive speaking or writing, especially the exploitation of figures of speech and other compositional techniques". On the other hand, *debate* is "a formal discussion on a particular matter in a public meeting or legislative assembly,

[1] Oxford English Dictionary [54]. Last consultation on 10th January 2021

in which opposing arguments are put forward and which usually ends with a vote". This means that, for both rhetoric and debate the goal is not to find an agreement, but to highlight the conflicting nature of the exchange. Conversely, as far as the dialectic is concerned, the aim is to collaborate with the interlocutor. This principle is essential for dialogue systems which should be able to co-operate with human users to be able to succeed in their task.

The connection between *dialectic* and *dialogue* becomes even clearer when considering its etymology. In fact, 'dialectic' derives from the Greek διαλεκτική (*dialektikí*), which means 'related to dialogue'. Philosophers described it as the process of two parties converging towards a shared view about a chosen topic. Socrates, for instance, referred to dialectic as the process of topic investigation based on questions and counterfactual evidences. In other words, he posed the basis of argumentation, which works with error detection. While his theory provides a means of verification of the truth about statements, it does not provide alternative solutions to confute statements or it does not explain why a certain statement cannot be accepted. On the other hand, Plato described dialectic as a method to find an absolute truth. Aristotle's dialectic is the one that has mostly influenced modern philosophy. He stressed the importance of the position expressed by a single individual, as opposed to the positions of a multitude of people. In modern philosophy, Hegel played a crucial role, as he used dialectic to to describe how the tension between a concept (i.e., *thesis*) and its opposite (i.e. *antithesis*) creates higher level concepts (i.e. *synthesis*). Since Hegel does not represent the dialectic as something related to communication but merely as a form of logic, Schopenhauer criticised his position. In fact, he offered a view of a dialectic which integrates human nature in the way dialogue can develop itself. In Schopenhauer's *The art of being right*, the competitive stance is crucial in dialectics, as dialogue strategies are seen as a means used not to verify the truth of the statements put forward by the opposing speaker but a means used to sway the opponent and elicit negative emotions.

For all these reasons, it can be stated that dialogue is originated through conflicts, which in turn are managed through dialectics. A dialogue system capable of managing informational inconsistencies must, therefore, apply dialectics, in that it has to be able to detect counter-arguments to the statements a user proposes to consider as shared knowledge. Modern dialogue systems, conversely, are often aimed at understanding the user's inputs rather than negotiating a common ground. Furthermore, when non-understanding or misunderstanding scenarios occur, clarification requests are mostly used to cover a recurring set of problematic contexts, such as when an acoustic or missing information (i.e. *slot-filling*) problem occurs. Nevertheless, specific forms of questions have always been considered as important dialectic strategies. In

Aristotle's *Topics*, it was pointed out that whereas polar questions can be used for dialectic premises, content questions cannot. In fact, content and alternative questions have a rhetoric function, whereas polar questions a dialectic one. For this reason, Aristotle's dialectical procedure relied exclusively on polar questions [96].

Dialectics in dialogue systems can be framed in the field of formal and computational argumentation, where two main research topics are listed: argumentation-based inference and argumentation-based dialogue. Argumentation-based inference concentrates on establishing what conclusions can be drawn starting from incomplete or inconsistent information. Argumentation-based inference models work similarly to Hegel's dialectic, since they investigate statements from a logical point of view without considering multiple participants. Historically, the first one who described an Abstract Argumentation Framework was Dung [58]. On the other hand, Pollock [117] first established the basis for formal argumentation-based inference. In his work, inference rules can be distinguished in *deductive* and *defeasible*[2] reasons. An argument can be attacked on the basis of its defeasible reasons either by attacking the conclusion of a defeasible inference by means of a conflicting conclusion or by attacking the inference itself without offering alternative solutions. Being based on arguments and *attack* relationships between arguments, inference graphs are used to graphically represent the structure over which conclusions can be drawn about posed statements.

Modern approaches to argumentation-based inference have used directed graph networks to generalise Dung's framework. For example, Abstract Dialectical Frameworks [25, 24] also rely on the concept of acceptance, according to which arguments can only be accepted when all conditions, or at least some of these, are accepted. This opens the way to a more general definition of *attack* conditions, and introduces other arguments, such as *support*. This representation also supports the choice of a tool like Neo4j which allows for knowledge representation in graph databases, as explained in Chapter 6.

Argumentation-based inference is different from argumentation-based dialogue, in that the former is a formal method which is applied to a single entity to decide about the truth of an argument. Therefore, this approach does not consider the problems arising from dialogues among different agents. In such cases, information is, in fact, distributed among the agents, who may or may not be willing to share it at different points in time due to individual strategies and goals. Before considering how argumentation-based dialogue works, we need to define, adopting a goal-oriented perspective, different dialogue cat-

[2] Defeasible reasoning refers to a rational compelling reasoning whose conclusions are not deductively drawn.

egories, as in [171, 173]:

- Persuasion: aimed at solving a difference of opinion;
- Negotiation: aimed at solving a conflict of interest by reaching a deal;
- Information seeking: aimed at information exchange;
- Deliberation: aimed at reaching a decision or at establishing a course of action;
- Inquiry: aimed at growth of knowledge and agreement *per se*;
- Quarrel: aimed at winning a verbal fight or a contest.

These classes, however, are not meant to be absolute, as multiple goals may be present during a single dialogue. Among the ones listed, persuasion dialogues appear to have been studied the most [179, 118].

Classical approaches to argumentation-based dialogue adopt the same setting that has been successfully used for argumentation-based inference, although their level of formalisation is inferior: inference rules are derived to establish a course of action. Structural relationships among *claims* and various kinds of *replies* are established in a formal protocol dedicated to establishing whether a speech act is legal or not. Since persuasion is the most studied situation in argumentation-based dialogue, a typical example of a formal communication language is the one described in [119]. In this type of setting, a *claim* provided by an agent A is supported by *data*, constituting an argument that can be explicitly put forward as a reply to a *why* move made by an agent B, which explicitly requests the speaker to explain the reasons why a statement should be accepted. Claims can be *attacked* by counter-arguments, which are other claims aimed at proving previous statements as false. *Conceding* and *retracting* moves, respectively, declare the acceptance of a statement or a change of attitude towards it, from commitment to non-commitment. Note that this does not imply a change of *belief*, as it is usually specified that the publicly declared position of an agent may not reflect what the agent actually believes.

Concerning deliberation dialogues, of interest for this work, the collaboration takes here place to find an optimal solution to a problem for which the involved agents have not yet a solution. For this type of dialogue, an interesting result was found. In case of a two-agents system adhering strictly to the communication protocol, forming their claims on the basis of their knowledge and adopting a collaborative attitude, it was demonstrated that the agreed solution is always acceptable to both parties [17]. This results from employing argumentation, whose usefulness in dialogue systems, designed for deliberation, was demonstrated in [84]. The problem that characterises argumentation-based dialogue with respect to argumentation-based inference is, therefore, the presence of different agents in the setting. This introduces multiple, not necessarily aligned, knowledge and, possibly, conflicting goals in the pursuit of a solution to a problem. Linguistic strategies adopted in such situations are of interest in this work. In conclusion, a solid argumentation-based dialogue theoretical framework is still missing because of the complexity of the phenomenon in question. In this work, a technological framework is proposed. This is based on different argumentation-based dialogue theoretical aspects, useful as a starting point to structure a linguistic-based theory which makes use of different declarative languages, such as Cypher and SPARQL, to be applied in conflicting deliberative human-machine dialogues (Chapter 6).

3.2 Corpus-driven Analysis for Linguistic Strategies in Argumentation

To understand which linguistic strategies are used as argumentation tools in deliberative dialogues when conflicts occur, a corpus analysis was needed at first. In fact, not only is the corpus analysis a test bench for annotation schemata, in that their coherence and applicability can be proved, but it is also important to investigate specific linguistic usages and to prove a theory right or wrong [93]. The starting point of the investigation of this work is the analysis of the German multimodal corpus SaGA. The goal of this analysis is, therefore, to investigate the forms and functions of CRs, following the proposed classification described in Chapter 2.

3.2.1 SaGA Corpus

The Bielefeld Speech and Gestures Alignment Corpus (SaGA) collects 25 multimodal dialogues of interlocutors which engage in a spatial communication task combining direction-giving and sight description [101]. Its aim is to document the use of speech and gestures to communicate information about the shape of objects and the spatial relations between them. Speakers with the role of *leader* were firstly exposed to a visual stimulus consisting in a virtual bus ride through a city, for which five specific landmarks (i.e. churches) had to be remembered. In the recording phase, the leader had to explain the route to another interlocutor with the role of *follower*. While describing the path, the leader had to be sure to have followed the route through the landmarks. Audio and video of each dialogue were recorded. As far as the videos are concerned, three different angles were considered: i) left, where the leader is; ii) right, where the follower is; iii) middle, where both interlocutors can be seen. In total, the corpus consists of 280 minutes of recorded video containing 4961 iconic/deictic gestures, 1000 discourse gestures, and 39,435 words. Concerning the transcriptions, utterances are broken into clauses associated with communicative goals, such as a) naming a landmark, b) landmark property description, c) landmark construction description, d) landmark position description. Furthermore, clauses were also distinguished in *theme*, the given information, and *rheme*, the new information, following Halliday [70]. Each clause was also divided in words, for which the corresponding lemmas and parts-of-speech were specified. No other speech-related pragmatic information were annotated. For this reason, CRs were identified and marked in ELAN, as far as, *problem*, *trigger*, *form*, *function* and *compromised item* labels were concerned (Chapter 2). For Common Ground CRs, as it will be further explained, information about original bias and contextual evidence are also taken into account. This corpus was chosen for two reasons: i) it is one of the largest and most comprehensive collections of naturalistic, yet controlled, systematically annotated speech-gesture data currently available, whose multimodal information can be used for future applications and investigations; ii) bias-evidence conflicts-derived polar questions, here correlated to the function expressed by Common Ground CRs, were investigated in German in available studies [56].

3.2.2 Quantitative and Qualitative Analyses

The SaGA corpus annotation and analysis brought to the collection of results which are the basis for answering RQ1 (see the Introduction), that is *which forms of clarification requests are frequently adopted by speakers when Common Ground Inconsistencies occur?* The corpus annotation was carried out by two annotators, namely an expert linguist and a computer science student. The levels considered in the annotation were *Communication Level*, *Trigger*, *Form*, and *Compromised Item*[3]. According to the Inter-Annotator Agreement scores, computed with the Cohen's Kappa [41], the agreement was substantial, as shown in table 3.1. Starting from the data collected in this analysis, the hypotheses which guided the studies carried out in this work are elaborated. In the 25 available dialogues of the SaGA corpus, 201 CRs were annotated, of which:

i) 51,7% are Missing Information CRs; the example 1 shows that the (omitted) subject of the CR is the same of the previous utterance for which more information are needed.

ii) 39,3% are Common Ground CRs; in the example 2 the negative form of a reduced polar question is used to correct a grounded information which

[3] The bias was not considered in the computation for its difficulty of interpretation. Its analysis needed, therefore, to be carried out with another experiment, as the one described in Chapter 4

Annotation Level	**Agreement**
Communication Level	1,0
Trigger	0,67
Form	0,78
Compromised Item	0,66
Agreement Mean Value	0,78 (Substantial)

Table 3.1: Inter-Annotator Agreement scores

is not confirmed by the contextual evidence, a condition which will be further investigated in the next Chapters.

iii) 3% are Noun Phrase Reference CRs; in the example 3 the alternative question is aimed at disambiguating the anaphoric elliptic noun phrase *Platz* (En. *square*) which can refer to two different previously mentioned locations.

iv) 3% are Deictic Reference CRs; the spatial reference *dazwischen* in the example 4 aims to be disambiguated with a CR in the form of a post-topic wh-question.

v) 1,5% are Meaning Ambiguity CRs; in the example 5 the meaning of the word *Spitze* is questioned via a CR in its reduced form.

vi) 1,5% are Unknown Meaning CRs; the example 6 serves to display the role of such CRs which question the meaning of a word, an expression, or a concept, such as *eckig U-förmig.*

The graph with the absolute frequency for each class is shown in Figure 3.2.

(1) a. *Also ist einfach ein großer Platz mit äh <SILENCE> ganz viel grau aufm Boden <SILENCE> ziemlich hässlich*

so it is simply a large square with uhm <SILENCE> a lot of gray on the ground <SILENCE> pretty ugly

b. *hat zwei Kirchentürme?*

does it have two bell towers?

(2) a. *da gehst dann weiter geradeaus <SILENCE> dann*

there then go straight <SILENCE> then

b. *da rechts nicht?*

not to the right?

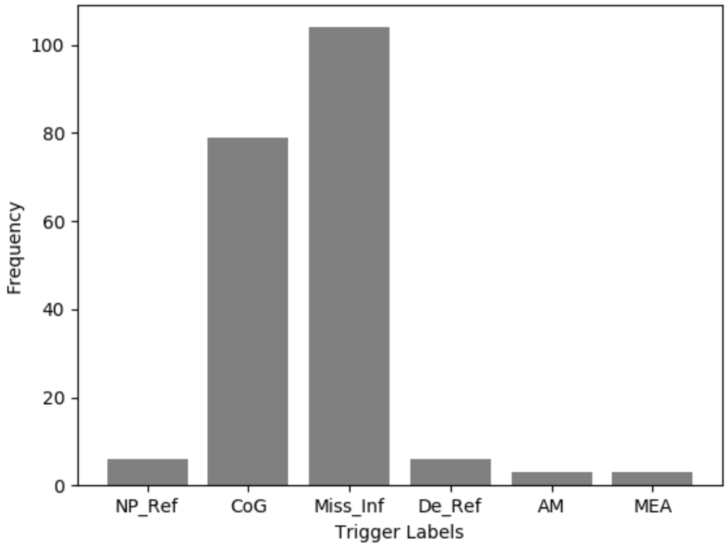


Figure 3.2: CR Triggers Distribution (Understanding Class)

c. *nein*

no

(3) a. *dann kommt man nicht zur Kirche dann kommt man zu diesem Platz*

at this point you do not arrive at the church but at this square

b. *ach so <SILENCE> Kirchplatz oder Rathausplatz?*

ah <SILENCE> the church square or the town hall square?

(4) a. *dazwischen war noch ein Platz <SILENCE> de war ganz nett ähm <SILENCE> da waren zwei Wandeltreppen*

in between there was another square <SILENCE> it was quite beautiful uhm <SILENCE> there were two spiral staircases

b. *wo zwischen jetzt?*

in between where now?

(5) a. *die hatte auch Spitze äh ähm wie heißt das*

it also had spikes uhm how do you say

b. *so so Zwiebeltürme?*

so bulb dome?

(6) a. *das Rathaus ist U-förmig <SILENCE> aber eckig nee <SILENCE> eckig aber U-förmig*

the town hall is U-shaped <SILENCE> but angular <SILENCE> angular but U-shaped

b. *eckig U-förmig?*

angular but U-shaped?

The annotated CRs all belong to the Understanding class, as Contact, Perception and Intention were not found. In fact, since interlocutors were asked to complete a task, the communicative channel was clearly open and the perception was constantly checked via back-channels. Furthermore, no intention-related problems were found, as the goal of the interaction was clear to the participants. Some understanding triggers were also not found, especially the syntactic-related ones, as the task was all in all not ambiguous. Information Processing CRs - Missing Information and Common Ground CRs - were the most frequent classes. This was related to the nature of the task, where checking the completeness and consistency of the received information was essential. Both classes occurred in different syntactic forms, as shown in Figure 3.3 and Table 3.2. Whereas Common Ground CRs are mostly uttered in the form of polar questions, Missing Information CRs are also formulated as alternative questions and wh-questions. The use of polar questions in this corpus confirms the tendency that, in task-oriented dialogues, getting 'hypotheses' confirmed (or, in our study, also checked) is preferred over asking for repetition [125]. In addition, since polar questions were defined as those types of questions that are more suitable to express a relatively high degree of epistemic stance, which in turn depends on the specific form of polar question (positive vs. negative), they resulted to be appropriate for grounding purposes which are important in this type of task.

As polar questions are seen as a bias vessel, the original bias and contextual evidence were also annotated in the corpus. Following Domaneschi [56], possible combinations between the original bias of the speaker and the contextual evidence are to be investigated, to point out the influence they may have on the choice of polar question forms. In [56], in fact, an experiment was carried out in German and English to retrieve information concerning the use of questions in specific proposed scenarios[4]. With the corpus analysis,

[4] Further details are given in Chapter 4

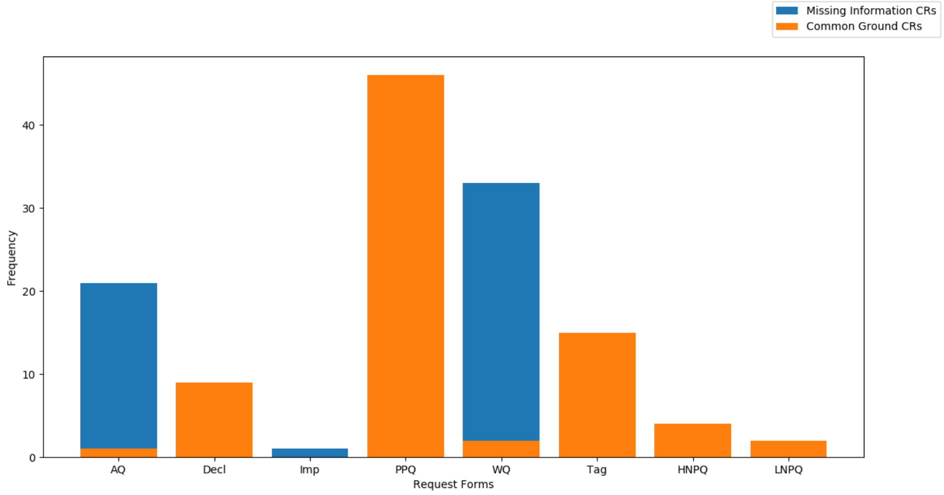


Figure 3.3: Requests' Form for Missing Information and Common Ground CRs - AQ: Alternative Questions; Dec: Declaratives; Imp: Imperatives; PPQ: Positive Polar Questions; WQ: Wh-Questions; Tag: Tag Questions; HNPQ: High Negation Polar Questions; LNPQ: Low Negation Polar Questions (see Chapter 2 for further details)

	Missing Information	**Common Ground**
WQ	34	2
AQ	21	1
PPQ	37	46
HNPQ	0	5
LNPQ	0	1
TPQ	8	15
Decl	3	9
Imp	1	0

Table 3.2: Question forms for Missing Information and Common Ground Clarification Requests

a comparison of language uses in similar situations could be carried out for German. Since annotators are not mind-readers, the original bias was annotated considering the answer of the interlocutor. Table 3.3 summarises the tendencies collected, whereas table 3.4 specifies the number of occurrences for each question form in each conflicting condition. Percentages of occurrences on the type of question and on the type of condition are shown, respectively, in Tables 3.5 and 3.6. Tendencies displayed in other studies are generally confirmed. Interestingly, negative polar questions are only triggered by problems in the Common Ground. Furthermore, TPQs are used exclusively in the positive form for Missing Information purposes, whereas in the negative form mostly for checking Common Ground consistency (12 occurrences out of 15). PPQs are the most frequent ones in both Missing Information and Common Ground CRs. Nonetheless, their frequency is lower in a specific condition, that is when a positive bias clashes with a negative evidence. In other words, when

		Bias		
		positive	**neutral**	**negative**
Evidence	**positive**	PQ	Decl/LNPQ/PPQ/TPQ	Decl
	neutral	HNPQ/PPQ/TPQ	Decl/PPQ/TPQ	
	negative	HNPQ/TPQ	Decl/LNPQ/PPQ	

Table 3.3: Results for preferred PQ form per pragmatic cell in SaGa Corpus

	HNPQ	**LNPQ**	**PPQ**	**TPQ**	**Decl**
positive-positive			7		
positive-neutral	1		1	9	
positive-negative	3			2	
neutral-positive	1		22	3	6
neutral-neutral			14	1	1
neutral-negative		1	2		1

Table 3.4: Occurrences for preferred PQ form per pragmatic cell in SaGa Corpus

a presupposition is confuted by an evidence, speakers tend to use a negative form rather than the positive one, expressing consequently a higher degree of conviction and of epistemic bias. This important result, which also confirms what observed in [56], will be further investigated in Italian, to understand if specific forms are also more appropriate in a Romance language in specific pragmatic situations, such as when Common Ground inconsistencies occur.

A further analysis shows that positive polar questions, which generally do not exhibit a 'bias' [56], mostly occur in the first phase of a dialogue (Figure 3.4) – when information in the PCG still has to be stored and for which one just needs confirmation – whereas negative polar questions, which are positively biased [56], occur more frequently in the last intervals of the interaction – when the receiver already has presuppositions based on the information stored in the PCG, which can be opposed to the negative contextual evidence of the previous turn. This observed tendency strengthened the hypothesis concerning the high appropriateness of negative forms to express a contrast in Common Ground inconsistencies scenarios.

In the next Chapters, the distribution of specific Common Ground CRs in the form of negative or positive polar questions will be further investigated. Specifically, polar questions used in different combinations of bias and evidence situations will be firstly analysed, as far as their appropriateness is concerned. Secondly, the analysis will be extended for human-machine applications in User Guided Tasks. This will allow for using specific Common Ground CRs to point out the internal state of the system through biased or unbiased hypothesis in order to help users to better solve understanding problems for a more natural interaction.

	HNPQ	LNPQ	PPQ	TPQ	Decl
positive-positive	0	0	0,152174	0	0
positive-neutral	0,2	0	0,021739	0,75	0
positive-negative	0,6	0	0	0,166667	0
neutral-positive	0,2	0	0,478261	0,25	0,666667
neutral-neutral	0	0	0,304348	0,083333	0,111111
neutral-negative	0	1	0,043478	0	0,111111

Table 3.5: Percentages on type of question for preferred PQ form per pragmatic cell in SaGa Corpus

	HNPQ	LNPQ	PPQ	TPQ	Decl
positive-positive	0	0	1	0	0
positive-neutral	0,090909	0	0,090909	0,818182	0
positive-negative	0,6	0	0	0,4	0
neutral-positive	0,03125	0	0,6875	0,09375	0,1875
neutral-neutral	0	0	0,875	0,0625	0,0625
neutral-negative	0	0,25	0,5	0	0,25

Table 3.6: Percentages on condition for preferred PQ form per pragmatic cell in SaGa Corpus

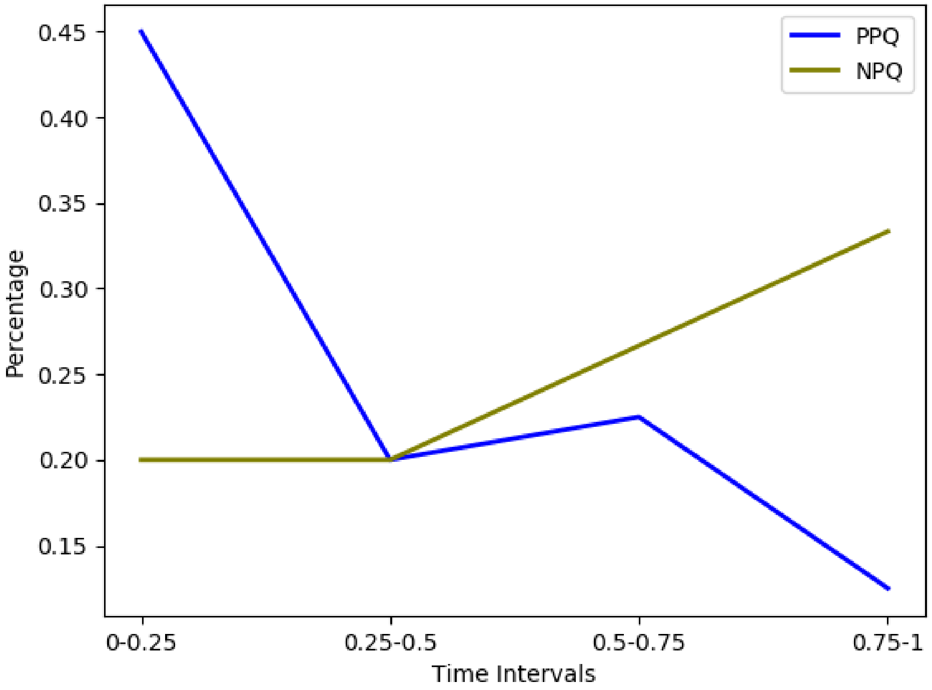


Figure 3.4: Distributions of Common Ground CRs (in the form of positive and negative polar questions) across the duration of the dialogues in the SaGA corpus [100].

Chapter 4

Linguistic Strategies in Common Ground Inconsistencies

In order to study which forms of polar questions are adopted to show conflicts in interaction, a first experiment was carried out. This study was modelled upon the one presented in [56] for English and German. The goal of the experiment was to check if specific forms of polar questions were also preferred in Italian when particular conflicts between an original bias and a contextual evidence occurred. This is important to understand if specific forms should be selected by a dialogue system when different types of conflicts arise in dialogue. In this section, the research hypothesis will be illustrated, along with the experimental setup and the collected results[1].

4.1 Polar Questions in Conflicting Representations

As explained in Chapter 2, polar questions usually encode in themselves not only a mere request but also presuppositions, agendas and preferences. Furthermore, when the questioner is closer to a K+ position, the use of a polar question can also implicate a disaffiliation. In this case, we refer to epistemically biased questions. According to the literature, one way of expressing disaffiliation is through the use of *Reversed Polarity Questions*, that are questions that convey bias towards the opposite valence than the utterance [85, 86]. For example, negative interrogatives can also function as positive assertions challenging the recipient's position [73]. Criticisms and challenges can also be expressed through declaratives (i.e. *You shouldn't have done that*), imperatives (i.e. *Don't do that to me again*), or exclamations (i.e. *How dare you?*), which are perceived more confrontational and explicit and can be there-

[1] The content of this chapter is also described in [51]

fore face-threatening [72, 149, p. 411]. Among non-standard communications, conflicting representations [76] are listed as interactions taking place when a discrepancy between what is communicated and what is believed by the agent occurs (Chapter 1). In these scenarios, polar questions can, therefore, serve as a knowledge challenging tool.

Different authors pointed out how either the original bias of the speaker or the contextual evidence bias could influence the syntactic form of polar questions.

Original speaker bias *Belief or expectation of the speaker that p is true, based on his epistemic state prior to the current situational context and conversational exchange* [91, p. 166].

Contextual evidence bias *Expectation that p is true (possibly contradicting a prior belief of the speaker) induced by evidence that has just become mutually available to the participants in the current discourse situation* [29, p. 7].

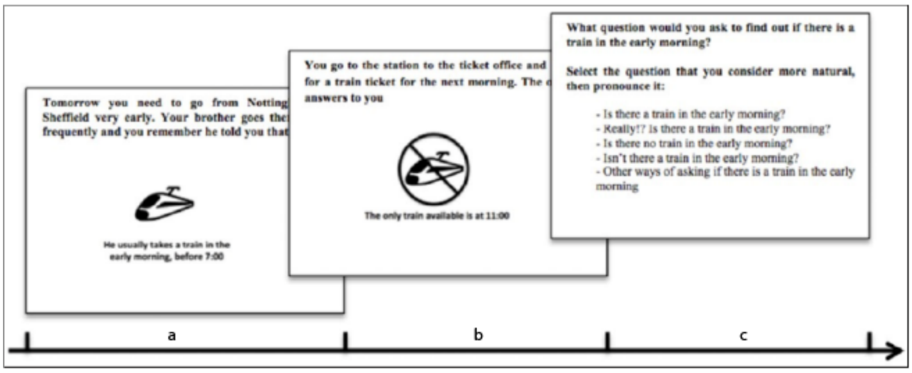
Following [56], possible combinations of the original bias of the speaker (where $B(p)$ is positive, B(-) is neutral, and $B(\neg p)$ is negative) and the contextual evidence (where $E(p)$ is positive, E(-) is neutral, and $E(\neg p)$ is negative) were investigated, in order to point out the influence they may have on the choice of polar question forms. This contrast represents, indeed, the conflict existing between the presupposed knowledge of the questioner and the one of the answerer. In section 4.1.2, the experiment carried out by the authors for English and German is considered as a starting point for the study presented in this work, whose aim is to check whether a pragmatic influence on polar questions' syntax also occurs in Italian.

4.1.1 Research Hypotheses

To answer RQ2 (see the Introduction), two different hypotheses guided the design of this first experiment.

H1 The bias-evidence conflict requires specific superficial polar question forms not only in not only in previously studied Germanic languages but also in the case of a Romance language, and more specifically in Italian, as in this study.

H2 Using a specific polar question form results in an improved communication efficiency, as the nature of the conflict can be, therefore, better signalised.



Tomorrow you need to go from Notting Sheffield very early. Your brother goes the frequently and you remember he told you that

He usually takes a train in the early morning, before 7:00

You go to the station to the ticket office and for a train ticket for the next morning. The answers to you

The only train available is at 11:00

What question would you ask to find out if there is a train in the early morning?

Select the question that you consider more natural, then pronounce it:

- Is there a train in the early morning?
- Really!? Is there a train in the early morning?
- Is there no train in the early morning?
- Isn't there a train in the early morning?
- Other ways of asking if there is a train in the early morning

Figure 4.1: Example of a trial for condition $B(p)\_E(\neg p)$, where 'a' builds the original bias, 'b' the contextual evidence, and 'c' shows the questions the participants had to choose from (Source: [56])

4.1.2 PolarExpress: Experimental setup

In Domaneschi et al. [56], the experiment consisted in a series of scenarios with six different types of conflicts randomly presented to participants. The scenarios presented ordinary fictional conversations, in form of dialogues made up of one or two turns (i.e., two friends preparing dinner, two students looking for the library). Each story was composed of two caption/picture pairs ('a' and 'b' in Figure 4.1), followed by the selection of the most appropriate PQ ('c' in Figure 4.1). Participants, therefore, had to choose one and only one appropriate question to pronounce. The choice was among five options: i) positive polar question (PPQ), ii) really-positive polar question (RPQ), iii) low negation polar question (LNPQ), iv) high negation polar question (HNPQ), v) and other. (Section 2).

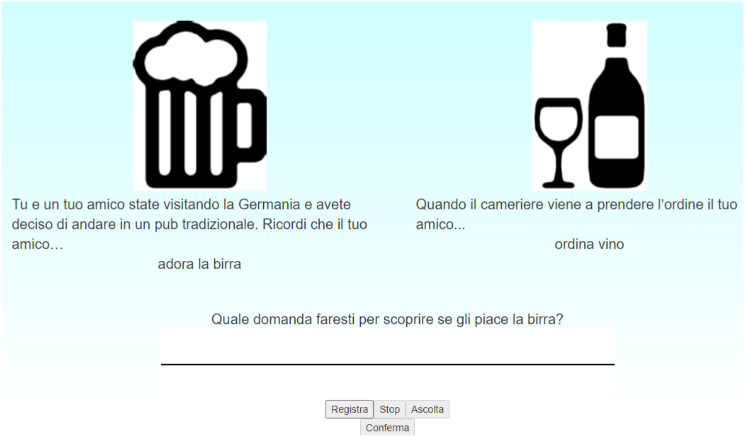
The first picture ('a' in Figure 4.1, on the left) aims at manipulating the original bias of the speaker; specifically, the utterance *He usually takes a train in the early morning before 7:00* is meant to generate a bias for the proposition p. On the other hand, the second illustration ('b' in Figure 4.1, in the middle) manipulates the bias triggered by the contextual evidence, as the utterance *The only train available is at 11:00* represents a negative evidence of the proposition p. The result of the reference study, in table 4.1, shows that both the original bias and the bias derived from the contextual evidence interact in the selection of the appropriate question: in both languages positive polar questions are typically selected when there is no original speaker belief and positive or non-informative contextual evidence is provided; low negation questions (i.e. *Do you not...?*) are most frequently chosen when no original belief meets negative contextual evidence; high negation questions (i.e., *Don't you...?*) are prompted when positive original speaker belief is followed by negative or non-informative contextual evidence; positive questions with *really*

		Original Bias		
		positive	**neutral**	**negative**
Contextual Evidence	**positive**		PPQ/RPQ	RPQ
	neutral	HNPQ (*outer*)	PPQ	
	negative	HNPQ (*outer/inner*)	LNPQ	

Table 4.1: Results for preferred PQ form per pragmatic cell in English and German (Adapted from: [56])

are produced most frequently when a negative original bias is combined with positive contextual evidence. Regarding HNPQ, we can distinguish two readings in the column with positive bias and neutral or negative evidence. Ladd [91] referred to the so-called *outer negation reading* when the speaker wants to double check p, and the *inner negation reading* in which the speaker wants to double check $\neg p$. In the inner reading, negation is part of the proposition being checked, whereas in the outer reading it is not. The two readings can be distinguished by the presence of positive polarity items (i.e. *some*, *already* or *too*), and negative polarity items (i.e. *any*, *yet*, *either*) [56].

Starting from these data, an extended version of the experiment was developed to collect similar tendencies in Italian. The collection was carried out online using a software specifically designed to administer the test. The 30 scenarios composing the data collection were selected from the English and German drafts used for the previous experiment in [56] and translated into Italian. The German data were preferred instead of the English ones, since the syntactic structures used in German were similar to the ones documented for the Italian language, specifically as far as the distinction between inner and outer reading for the high negation polar questions. This will be specified in detail in section 4.2.2. The pragmatic situations driven by the combination of original bias and contextual evidence which were collected are $B(p)\_E(p)$, $B(p)\_E(\text{-})$, $B(p)\_E(\neg p)$, $B(\text{-})\_E(p)$, $B(\text{-})\_E(\text{-})$, $B(\text{-})\_E(\neg p)$, $B(\neg p)\_E(p)$, $B(\neg p)\_E(\text{-})$, $B(\neg p)\_E(\neg p)$. Nevertheless, $B(p)\_E(p)$, $B(\neg p)\_E(\text{-})$, and $B(\neg p)\_E(\neg p)$ were left out from the analysis. In fact, as pointed out in [56], speakers with an original bias for p that receive contextual evidence for p will assume that p is true and will not question further about the its truth. Similarly, the same happens for the $B(\neg p)\_E(\neg p)$ condition. In [128], polar questions were rated as not natural in the aforementioned conditions. As far as the $B(\neg p)\_E(\text{-})$ condition is concerned, the appropriate polar questions described in [129, 6] are a combination of high and low negation. In fact, these two forms also resulted to be frequently selected as appropriate in the present study. Nevertheless, these three conditions were left to future analysis, in order to focus on conditions which were more suitable for the description of conflicting scenarios. An exception is made as far as the first task of this study



Tu e un tuo amico state visitando la Germania e avete deciso di andare in un pub tradizionale. Ricordi che il tuo amico…

adora la birra

Quando il cameriere viene a prendere l'ordine il tuo amico...

ordina vino

Quale domanda faresti per scoprire se gli piace la birra?

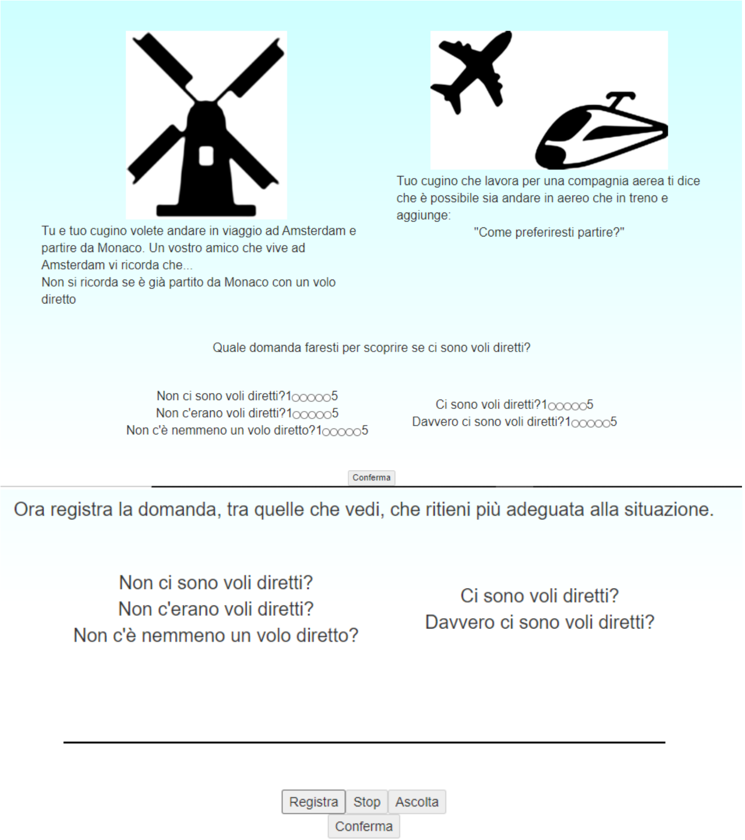
Figure 4.2: [Free Production Task] Original bias: *You and a friend of yours are visiting Germany and have decided you want to go eating in a traditional pub. You remember that your friend loves beer* (positive); Contextual Evidence: *When the waiter comes to take your orders, your friend orders wine.* (neutral)

is concerned (Free Production), for which also other forms were collected, such as wh-questions, which might be considered as more appropriate than polar questions in these conditions. Further details are given in section 4.2.

The target subjects were limited to the Campania region, in order to avoid the diatopic variation to influence the choice. In fact, to control the regional variety and to ensure the gender balance, each participant had to firstly answer a sociolinguistic questionnaire, concerning age, gender (male, female, and other), geographical origin, other places where they lived more than 12 months, and other spoken languages. To ensure that each possible bias-evidence combination for each task occurred, 81 participants were needed. The resulting sample comprises 41 females, 39 males, 1 other, with an average age of 32,37.

Each participant was provided with 10 different scenarios. For each of them, they were asked to perform one, randomly selected, of the three different planned tasks. In fact, contrary to what established in Domaneschi et al. [56], three different tasks were here randomly shown. Furthermore, for two of the three tasks, instead of asking them to only select one form, as in [56], they could evaluate their appropriateness reflecting the natural tendency of speakers to use more than one form to express the same function. The tasks are described in detail below:

A Free Production (FP): participants were asked to spontaneously record a question in order to acquire a specific piece of information for that particular situation (Figure 4.2). This additional type of task is useful to collect information concerning the spontaneous choice of question types depending on pragmatic needs. Furthermore, the intonational patters that could be extracted from such spontaneous choices can be adopted



Tu e tuo cugino volete andare in viaggio ad Amsterdam e partire da Monaco. Un vostro amico che vive ad Amsterdam vi ricorda che...
Non si ricorda se è già partito da Monaco con un volo diretto

Tuo cugino che lavora per una compagnia aerea ti dice che è possibile sia andare in aereo che in treno e aggiunge:
"Come preferiresti partire?"

Quale domanda faresti per scoprire se ci sono voli diretti?

Non ci sono voli diretti?1○○○○○5
Non c'erano voli diretti?1○○○○○5
Non c'è nemmeno un volo diretto?1○○○○○5

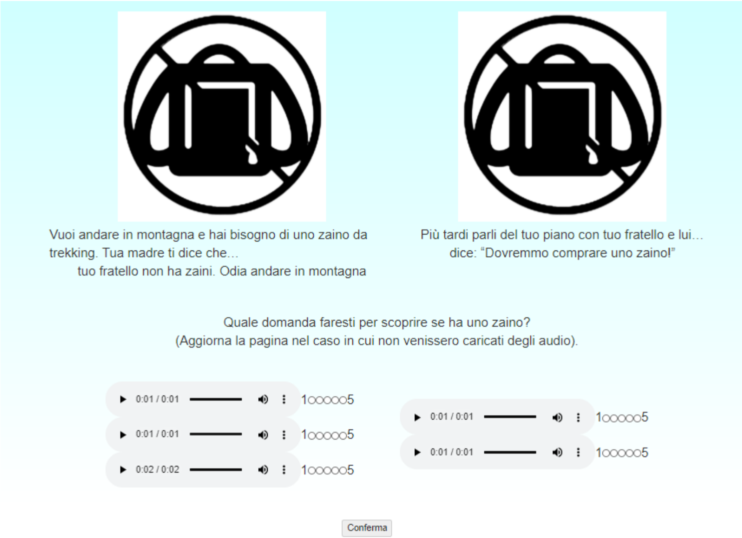
Ci sono voli diretti?1○○○○○5
Davvero ci sono voli diretti?1○○○○○5

Ora registra la domanda, tra quelle che vedi, che ritieni più adeguata alla situazione.

Non ci sono voli diretti?
Non c'erano voli diretti?
Non c'è nemmeno un volo diretto?

Ci sono voli diretti?
Davvero ci sono voli diretti?

Figure 4.3: [Guided Production Task] Original bias: *You and your cousin want to travel from Munich to Amsterdam. A friend of yours who lives in Amsterdam tells you that he does not remember if there is a direct flight* (neutral); Contextual Evidence: *Your cousin who works for a flight company tells you that it is possible to travel both by flight and by train and ask you "How do you want to travel?"* (positive)



Vuoi andare in montagna e hai bisogno di uno zaino da trekking. Tua madre ti dice che...
tuo fratello non ha zaini. Odia andare in montagna

Più tardi parli del tuo piano con tuo fratello e lui...
dice: "Dovremmo comprare uno zaino!"

Quale domanda faresti per scoprire se ha uno zaino?
(Aggiorna la pagina nel caso in cui non venissero caricati degli audio).

Figure 4.4: [Synthesis Scoring Task] Original bias: *You want to go to the mountains and you need a hiking backpack. Your mother tells you that your brother does not have any backpack. He hates to go to the mountains* (negative); Contextual evidence: *Later, you talk with your brother about your plan and he says: "We should buy a backpack"* (negative)

for the definition of prosody-pragmatics interface schema.

B Guided Production (GP): participants were provided with a set of different written forms of polar questions, for each of which they have to give a score from 1 to 5, according to their appropriateness in that determined situation (where 1 corresponds to a question completely inappropriate and 5 to completely appropriate). Once having rated the questions, participants also had to record the one they considered to be the most appropriate (Figure 4.3). In this way, the spoken production of the selected questions is also collected.

C Synthesis Scoring (SS): five synthesised polar questions were reproduced, for each of which the participants have to give a score from 1 to 5, according to their appropriateness (Figure 4.4). The questions were synthesised via neural text-to-speech services provided by Microsoft, whose intonation is based on statistical patterns extracted from training data. This is important considering the lack of described intonational schema for bias-evidence contrast in Italian polar questions. In fact, the selected patterns are here considered as a starting point with the aim of understanding if some frequent patterns are generally adequate to express a particular type of conflict.

For GP and SS tasks, the question forms provided were based on the ones selected in [56]. Five stimuli were therefore presented. Contrary to the previous experiment, the option *other* was left out, as the participants had the possibility to assign low scores to all the proposed items, if none was considered appropriate. Since no stimulus is considered appropriate in a situation, others might be a better option for the user. Furthermore, the possibility to consider other syntactic forms rather than polar questions as appropriate in some situations was also inferred by the FP task. The option *other* was, therefore, substituted with a high negation polar question in the past tense. This choice lies on empirical considerations. In fact, this form seems to be more frequently adopted and seems to convey a stronger degree of the speaker's bias. Note that in [56], changes in tense, word order, and addition of particles were ignored if they did not affect the biases at issue.

4.2 Analysis and Results

In this section, the data gathered during the experiment are presented and analysed for each of the tasks carried out.

4.2.1 Free Production

The FP task was aimed at collecting spontaneous productions from the participants. They were, therefore, asked to record the most appropriate question in the presented situations without giving them possible options among which to choose. As reported in figure 4.5, HNPQs and HNPQ_Ps were more frequently chosen in $B(p)\_E(\neg p)$ and $B(\text{-})\_E(\neg p)$ situations. On the other hand, LNPQs were also more frequently selected in $B(\text{-})\_E(\neg p)$ situations, but in smaller numbers compared to HNPQs and HNPQ_Ps. HNPQs were also frequent in $B(\text{-})\_E(\text{-})$ situations but not as much as PPQs. In fact, PPQs, for their versatility, were produced in $B(p)\_E(\text{-})$, $B(\text{-})\_E(p)$, $B(\text{-})\_E(\text{-})$, and $B(\neg p)\_E(p)$ situations. RPQs were produced exclusively in $B(\text{-})\_E(p)$ and $B(\neg p)\_E(p)$ situations, as in [56].

Since participants were free to record any stimulus they considered appropriate, wh-questions were also produced. Interestingly, these forms mostly appear in pragmatic conditions, where the speaker has no original bias against positive (7), neutral (8), or negative evidence (9). One possible interpretation for this choice can refer to the fact that in some cases the bias had a major impact on the speakers, bringing them to collect additional information in case of lack of knowledge. On the other hand, the frequent selection of wh-questions in $B(\neg p)\_E(p)$ scenarios might be due to a major impact of the evidence on the speaker. In fact, instead of asking confirmation with an epistemic adverb like *really*, as expected, speakers might rely on the negative evidence and inquire more about it. Another interesting results is given by the use of such questions in $B(p)\_E(p)$ and $E(\neg p)\_E(\neg p)$ situations, which were left out in the resulting analysis (and in the reference graph), as explained in (Section 4.1.2). This choice can explain the alleged inappropriateness of polar questions in those scenarios.

(7) Come faccio a fare la tessera?
En. *How can I get the badge?*

(8) A quale negozio stai pensando?
En. *What shop are you thinking about?*

(9) Quanto è lontano il supermercato?
En. *How far is the supermarket?*

Furthermore, the standard polar question forms considered in the other tasks of the experiment were in few cases also enriched with other linguistic markers used to convey different degrees of bias, as shown in table 4.2.

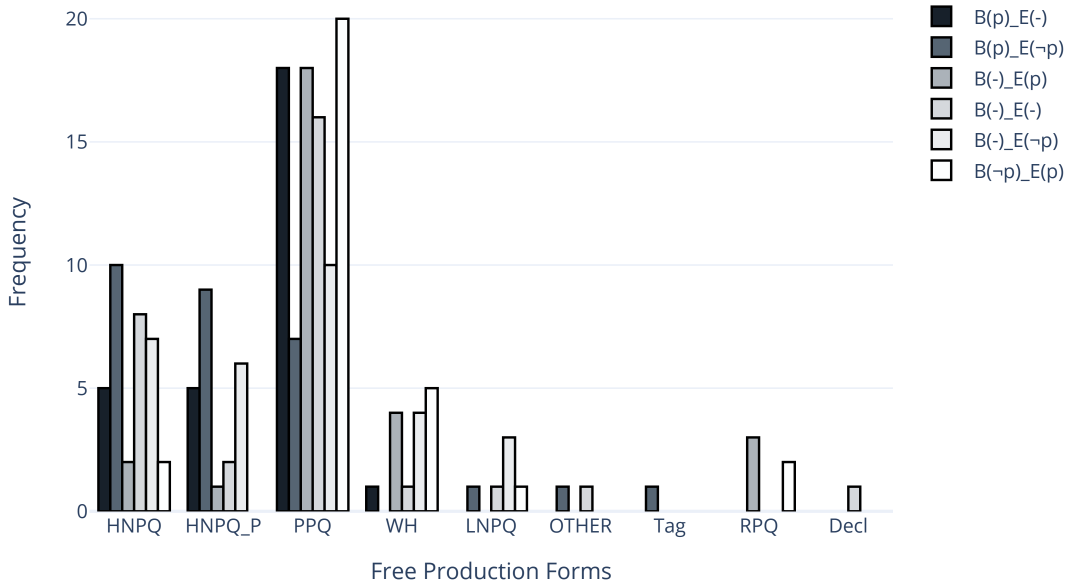


Figure 4.5: Free Production Results

In fact, as also reported in [104], there are different types of bias which are linked to their illocutionary force. Specifically, we can differentiate between: i) epistemic bias, reflecting what the speaker thinks, expects, or knows the right answer is; ii) deontic bias, reflecting what the speaker judges the right answer ought to be; iii) desiderative bias, what the speaker wants the right answer to be. For example, it is interesting to point out that HNPQs, especially in the past tense, which are mostly used in $B(p)\_E(\neg p)$ situations, can be preceded by the adversative conjunction marker *ma* (En. *but*). This marker is, indeed, used to *question the correctness of a new, adversative or contrasting referent, circumstance, or situation* [107]. Facing this contrasting contextual evidence, the speaker needs, therefore, to express strongly its *hope*, as defined in [104], toward the correctness of their presupposition. In this case, the conjunction is used to express an epistemic bias. Interestingly, the strength of this epistemic marker is used exclusively in combination with HNPQ_Ps (44% of the HNPQ_Ps were preceded by *ma*) whose adequateness in $B(p)\_E(\neg p)$ was proved to be unquestionable, as also shown in the next tasks of this experiment.

On the other hand, the adversative conjunction is less frequently used with HNPQs and not used at all with PPQs. These forms were, conversely, sometimes used with other types of epistemic markers. These can be described as part of what is called 'epistemic modality'. Epistemic modality refers to a *conjecture about the truth value of a proposition* [107]. This is used in questions expressing a supposition interpretable either as a statement or a question depending on the epistemic status of the speaker and the listener [107]. For example, in **??**, the marker *forse* (En. *maybe*) is used in combination with a PPQs in a B(-)_$E(p)$ condition to express an epistemic possibility. In 11, on the other hand, an epistemic expression introducing the HNPQ is used to express doubts towards the given evidence. Moreover, PPQs were frequently used in combination with the causal conjunction *quindi* (En. *so*), as in example 12. As also described for the Spanish language gomez1993conectores, this conjunction is used with the conversational role corresponding to confirmation request. In fact, PPQs of this type were mostly used when this function was needed (B(-)_$E(p)$, $B(p)$_$E(p)$, and $B(\neg p)$_$E(p)$). PPQ_implicit, on the other hand, refers to PPQs which were preceded by other phrasal expressions, as in 13, where the pragmatic function is of information-seeking.

(10) Forse hai l'assicurazione?
En. *Do you maybe have an insurance?*

(11) Sei sicuro che non c'è un negozio di elettronica?
En. *Are you sure there isn't an electronic store?*

(12) Quindi c'è una biblioteca universitaria?
En. *So is there a university library?*

(13) Sai se c'è un ristorante?
En. *Do you know if there is a restaurant?*

These alternative forms, representing a lower percentage of participants' choices, were not deepened in this work.

4.2.2 Guided Production

As far as the guided production task is concerned, the data analysis regards on the one hand the scores and on the other the selection of one of the items to be pronounced. The results representing the speakers' tendencies in evaluating the appropriateness of specific question forms according to the type of conflict are summarised in figure 4.6. Here, the percentages of the highest scores for each question type in each conflict situation are shown. The statistical analysis were carried out with R [160]. The data were firstly analysed with

Label	**Occurrence**
HNPQ	80,39%
HNPQ_ma	1,92%
HNPQ_sicuro	7,69%
HNPQ_possibile	1,92%
HNPQ_quindi	7,69%
HNPQ_vero	1,92

Label	**Occurrence**
HNPQ_P	55.56%
HNPQ_P_ma	44,44%

Label	**Occurrence**
PPQ	81,39%
PPQ_P	0,77%
PPQ_quindi	9,30%
PPQ_perché	0,77%
PPQ_implicit	3,87%
PPQ_sicuro	1,55%
PPQ_ancora	0,77%
PPQ_forse	1,55%

Table 4.2: Percentage of PQs with and without epistemic markers in the FP task

the Shapiro-Wilk normality test [147] to check distributional assumptions. In all combinations of bias and evidence, at least one form had a non-normal distribution of the scores, so non-parametric tests were used. To compare the mean values of the distributions, the Kruskal–Wallis test [89] was used to check the existence of significant differences. In all cases, the test indicated that at least one significant difference was present; these were further detailed using the pair-wise Wilcoxon test [177]. The H_0 states that there is no statistically significant difference among the average values of the analysed distributions. More specifically, the probability that the observed difference is due to chance is endorsed in the H_0. The rejection of the H_0 would, therefore, mean that the difference is statistically significant. The practical interpretation in this study would be the preference for one question form in each situation. Conflict-related results are going to be described and discussed in detail in the next sections.

Positive Bias vs. Neutral Evidence For the $B(p)\_E(\text{-})$ conflicts, PPQs, HNPQs, and HNPQ_Ps show the highest scores (Figure 4.6), where in [56] HNPQs were selected. The data presented in figure 4.7 and table 4.3 confirm this tendency with respect to LNPQs and RPQs, as they are not perceived as appropriate in this situation: they are chosen less frequently in a statistically significant way when compared with PPQs, HNPQs, and HNPQ_Ps. Differently from [56], PPQs appear to be a valid choice, since no statistically

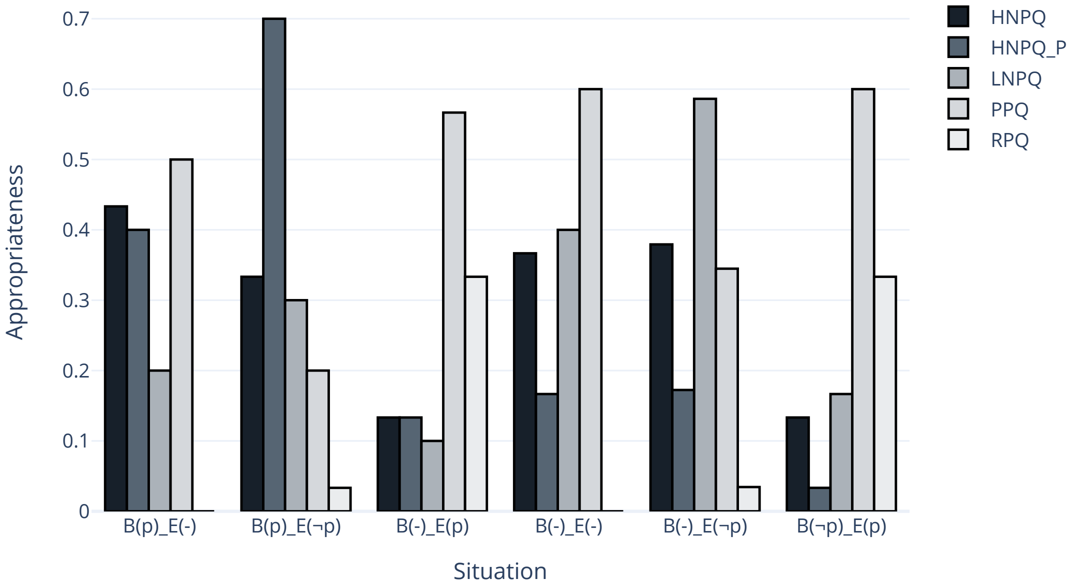


Figure 4.6: Percentage of highest scores for each type of polar questions in different situations

significant difference is found between the three question types. This can be explained by the fact that, according to the way the question is pronounced, PPQs can actually be preferred, because they can show the same pragmatic function and, at the same time, do not damage the *face* [67] of the interlocutor. In fact, the explicit reference to the conflict through the use of a negation can represent a threat, especially in a situation where the evidence is perceived to be not strong enough (i.e. *neutral*).

Positive Bias vs. Negative Evidence For the conflict arising from a strong presupposition and an evidence contradicting it, HNPQ_Ps are scored as more appropriate (Figure 4.6). The Box plot in figure 4.7 and the table 4.3 show that this tendency has strong statistical significance when its appropriateness is compared with that of PPQ and RPQ. Conversely, significance is lost when compared to LNPQ (p = 0.08). Interestingly, this conflict type was defined as the *ambiguity cell* in [56], as far as the English data were concerned. This ambiguity derives from the fact that, in English, HNPQs can have an inner or outer reading. The difference between inner and outer HNPQs depends on the polarity of the proposition being checked. In fact, in the inner reading, the negation is part of the proposition being checked (question about a negative proposition), whereas in the outer reading it is not (question about an affir-

		HNPQ	**HNPQ_P**	**LNPQ**	**PPQ**
$B(p)\_E(-)$	**HNPQ_P**	x	-	-	-
	LNPQ	**	x	-	-
	PPQ	x	x	*	-
	RPQ	**	**	**	**
$B(p)\_E(\neg p)$	**HNPQ_P**	x	-	-	-
	LNPQ	x	x	-	-
	PPQ	x	**	x	-
	RPQ	**	**	**	**
$B(-)\_E(p)$	**HNPQ_P**	x	-	-	-
	LNPQ	x	x	-	-
	PPQ	*	**	**	-
	RPQ	x	x	x	*
$B(-)\_E(-)$	**HNPQ_P**	*	-	-	-
	LNPQ	x	x	-	-
	PPQ	x	*	x	-
	RPQ	**	**	**	**
$B(-)\_E(\neg p)$	**HNPQ_P**	*	-	-	-
	LNPQ	x	** -	-	-
	PPQ	x	x	*	-
	RPQ	**	**	**	**
$B(\neg p)\_E(p)$	**HNPQ_P**	x	-	-	-
	LNPQ	x	x	-	-
	PPQ	**	**	**	-
	RPQ	x	**	x	x

Table 4.3: Statistically significant differences in different pragamtic situations. No significance is marked with x ($p > 0.05$); weak significance is marked with * ($0.01 < p < 0.05$); strong significance is marked with ** ($p < 0.01$)

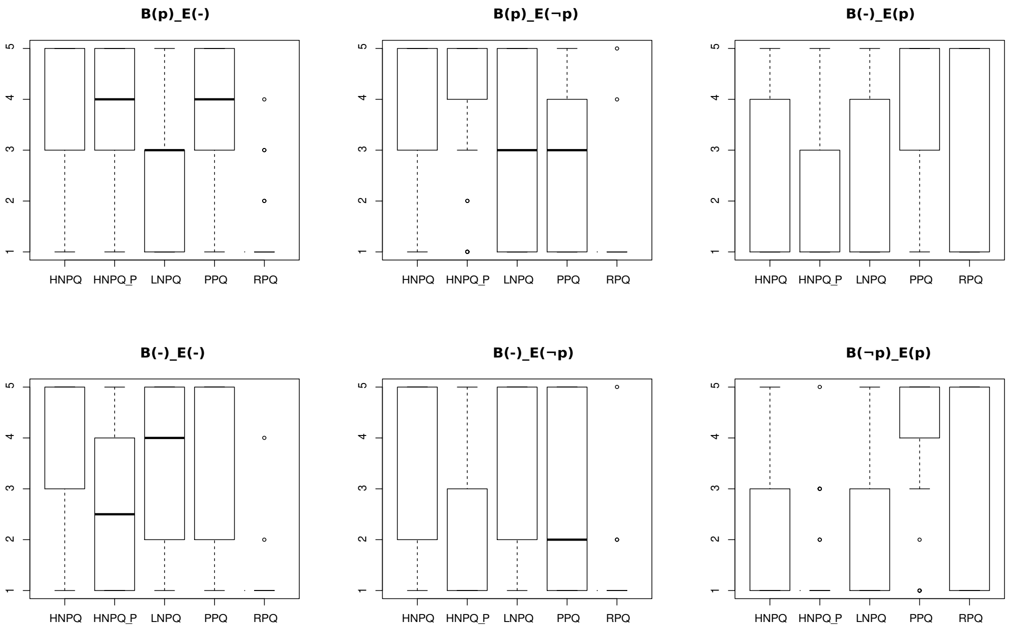


Figure 4.7: Boxplots showing scorse for Polar Questions forms in different pragmatic conditions

mative proposition [91]). This means that, with the outer reading the original belief p is double-checked (i.e., *Isn't there some good restaurant around here?*), whereas with the inner reading the opposite proposition ($\neg p$) is double-checked (i.e., *Isn't there any good restaurant around here?*). The English data in [56] show that for the $p/\neg p$ condition both inner and outer readings are possible. In German, the difference between HNPQ and LNPQ in this situation is lower, since the pragmatic meanings of inner HNPQs and LNPQs are similar. In fact, in German and in Italian, inner HNPQs have the same form as LNPQs, and both readings are possible. This can be the explanation for a lack of a statistically significant difference in the HNPQ/LNPQ situation ($p = 0.35$) and in the HNPQ_P/LNPQ situation ($p = 0.08$), for Italian. Furthermore, although HNPQ_Ps are preferred in $B(p)\_E(\neg p)$ conditions, the difference between the past tense and present tense in the negation does not lead to a strong refutation of the H_0 ($p = 0.35$). This confirms what has been described in [56], where the high negation was preferred with a percentage of 67%, although the authors did not take into account the tense.

Neutral Bias vs. Positive Evidence In situations where there is no original bias combined with positive evidence, PPQs are considered to be more appropriate (Figure 4.6), as also demonstrated in [56]. In fact, their appropriateness is statistical significant when compared with that of the negative polar questions (Figure 4.7; Table 4.3). The statistically significant difference with RPQs is, instead, lower ($p = 0.03$). In English and German, a similar, but slightly stronger, tendency was noted [56]. In fact, the preposed *really* was supposedly interpreted as a discourse particle with the function of expressing interest and engagement and not as an epistemic adverb asking for confirmation about the proposition, as expected for the negative-positive scenario.

Neutral Bias vs. Neutral Evidence When neither original bias nor contextual bias are displayed, PPQs are preferred around 60% of the time, as in English and German [56]. A weak statistically significant difference is shown when PPQs are compared with HNPQ_P ($p = 0.02$) as shown in figure 4.7 and table 4.3. No statistically significant difference, instead, occurred between PPQs and HNPQs/LNPQs ($p = F0.9$). In fact, as hypothesised in [56], HNPQs can be used in this situation when only the contextual evidence is considered, whereas LNPQs are selected when only the original bias is considered.

Neutral Bias vs. Negative Evidence In $B(\text{-})\_E(\neg p)$ conflicts, LNPQs are preferred as for English and German [56], with a statistically significant difference detected only when compared to HNPQ_Ps (Figure 4.7; Table 4.3). The comparison with the negative polar questions follows the same explanation reported for the previous conflict. Furthermore, this scenario was also problematic, as the mention of the p-proposition to question about was perceived as unexpected for the participants because it was already negated by the evidence.

Negative Bias vs. Positive Evidence Contrary to what was expected and discussed in [56], in this conflict scenario, PPQs were considered to be more appropriate than RPQs (around 60%). As reported in figure 4.7 and table 4.3, PPQs are preferred with statistically significant difference with respect to HNPQs, HNPQ_Ps and LNPQs. There is no statistically significant difference with RPQs, as the preposed *really* was supposedly interpreted as an epistemic adverb with a confirmation function, as expected. One possible explanation of the highest preference for PPQs can be found in their production. In fact, both RPQs and PPQs produced with an accent on finite verbs can be used with a negative original bias for confirmation purposes [7]. This use of the RPQs is described in Chapter 4.3.

In the second part of the GP task, participants were asked to choose only one of the options to be recorded. Almost all the tendencies that were reported for the first part were confirmed, as shown in figure 4.8. Only for the $B(p)\_E(\text{-})$ and the $B(\neg p)\_E(p)$ situations the tendencies are slightly different. In the scoring part of the experiment, for the $B(p)\_E(\text{-})$ situation the PPQs were

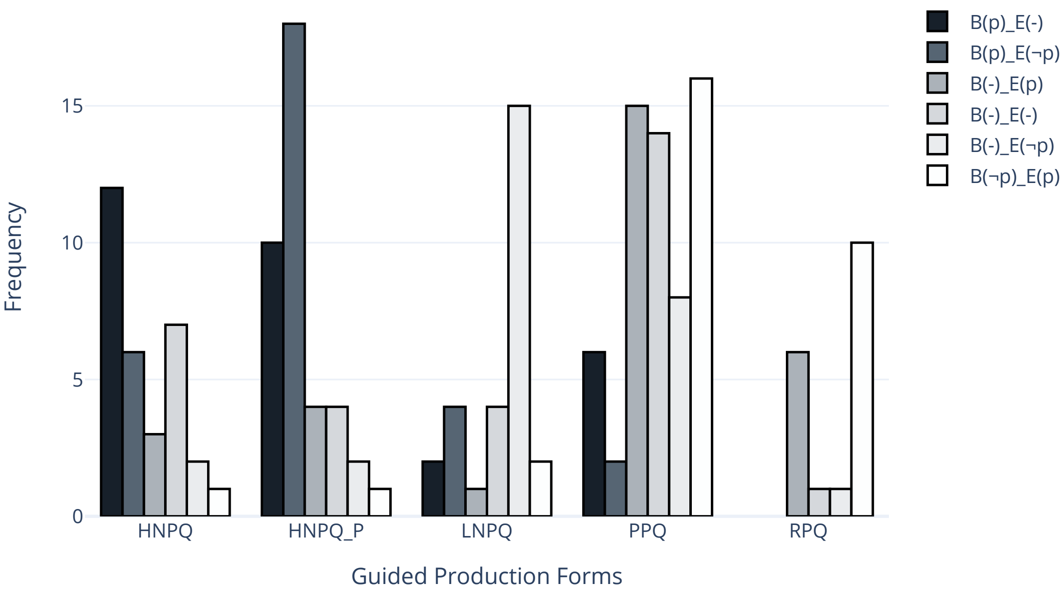


Figure 4.8: Guided Production Results - Selection

rated as the most appropriate, although there was no statistically significant difference from HNPQs and HNPQ_Ps. Here, HNPQs are more frequently chosen and PPQs are chosen right after them, as in [56]. Similarly as in [56], for the $B(\neg p)\_E(p)$ situation, the RPQs were more frequently selected, where in the first part the PPQs were rated as more frequently as more appropriate. In conclusion, it can be stated that positive polar questions are considered to be more versatile and generally more appropriate in non-conflicting scenarios, whereas negative polar questions - high, low, or in the past tense - are more appropriate when different kinds of conflict occurs in the contextual evidence.

4.2.3 Synthesis Scoring

Regarding the synthesis scoring task, the data analysis is concerned with the scores participants gave to each one of the given option. The results representing the speakers' tendencies in evaluating the appropriateness of specific question intonational forms according to the type of conflict are summarised in figure 4.9. As in the GP task, HNPQ_P collected higher scores in $B(p)\_E(\neg p)$ scenarios. The same form beated the others in the $B(p)\_E(\text{-})$ scenario, where in the GP task the PPQ had the highest score. This could be explained by the fact that the written form can be interpreted differently, whereas the synthe-

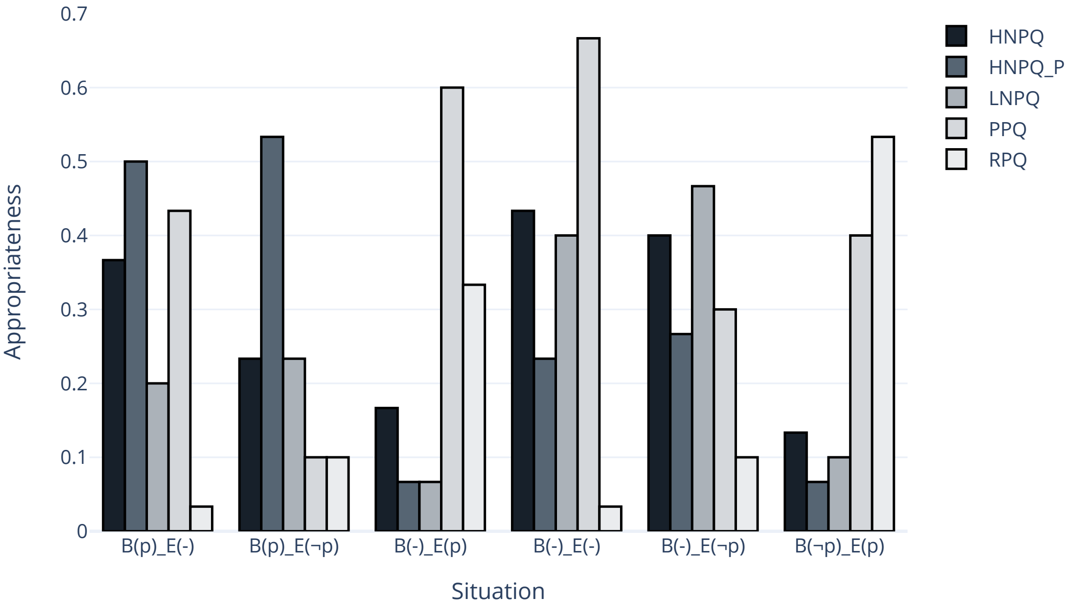


Figure 4.9: Synthesis Scoring Results

sised forms have fewer perceived possible interpretations. The PPQ is generally preferred in $B(\text{-})\_E(p)$ and $B(\text{-})\_E(\text{-})$ scenarios, whereas the LNPQ is preferred in $B(\text{-})\_E(\neg p)$ scenarios, similarly to the GP task. Differently from the previous task, the RPQ is here preferred in $B(\neg p)\_E(p)$ conflicts. The collection of such results will be used as a term of comparison for the productions that seem to be far from the standard. In fact, we are not yet able to tell which intonational patterns are typical of specific polar questions in Italian. One possible future application for these results can be, in fact, the analysis of the deviating forms which could have been chosen by the participants to communicate specific pragmatic meanings.

4.3 Preliminary Prosodic Analysis

Since some syntactic forms in the different experimental scenarios were rated as appropriate in different pragmatic situations, it can be hypothesised that other levels of analysis can interfere with the pragmatic sphere. As mentioned in Chapter 2, epistemic gradients, and therefore biases towards specific presuppositions, can be expressed via syntax and/or prosody. Specifically as far as polar questions are concerned, it was studied how the prosodic realisation could influence their pragmatic interpretation. According to Savino & Grice

[135], who analysed polar questions in the Italian variation of Bari, it was pointed out that the pitch accent type determines the difference between neutral question and check. Moreover, in was stated that a greater pitch span in the pitch accent was generally used to express a negative bias. In [112], on the other hand, the perception analysis highlighted that both phonological (i.e., position and type of nuclear pitch accent) and phonetic (i.e., pitch span) phenomena are used by Salerno listeners to recognise the bias of a polar question.

Despite not being the main goal of this work, the data collected through FP and GP tasks were qualitatively analysed to look for tendencies, especially for those forms which were used to express various combinations of bias-evidence. To pursue this aim, a bottom-up approach was followed, as we started from observations to find patterns and possible hypotheses. To do that, the audios were transcribed and TextGrid files containing word boundaries, syllables, and phonetic transcriptions were generated using WebMAUS [83]. The analyses were carried out in Praat [19]

PPQ As already pointed out, PPQs were the most frequent forms because of their versatility. Because of their application in different pragmatic situations, and because we are aware of the mutual influence of prosody and pragmatics, it can be hypothesised that prosodic differences can be observed in each situation. Nevertheless, since the aim of the experiment was to collect syntactic forms rather than intonation patterns, not enough data are now available to prove this hypothesis, which needs, therefore, further investigations for future work. For merely description purposes, pitch contours of PPQs in $E(\text{-})\_E(\text{-})$ scenarios are shown in figure 4.10, as this is the situation in which this form was mostly adopted, despite with no statistical significance with respect to the other question forms. Such tendency, in fact, can show an impact on the pragmatic-related intonation used across syntactic realisations. In general, PPQ are realised with a rising or rising-falling f_0 movement within the stressed syllable of the word or phrase bearing the nuclear pitch accent. The nuclear prominence was either found to be realised on the sentence-initial verbal phrase or on the following noun phrase. Secondary prominences might be detected in case of long utterances, such as those containing adverbials or prepositional phrases modifying the noun phrase. Finally, PPQ could either end with a low or with a high boundary. In the data collected, the boundary tone could be both high, as in the first example, *Hai una bici?* (En. *Do you have a bike?*), in Figure 4.10, or low, ad in the other two examples *C'è un pullman dopo le 21?* (En. *Is there a bus after 9pm?*), *C'è una metro qui vicino?* (En. *Is there a metro station nearby?*), in the same figure. As far

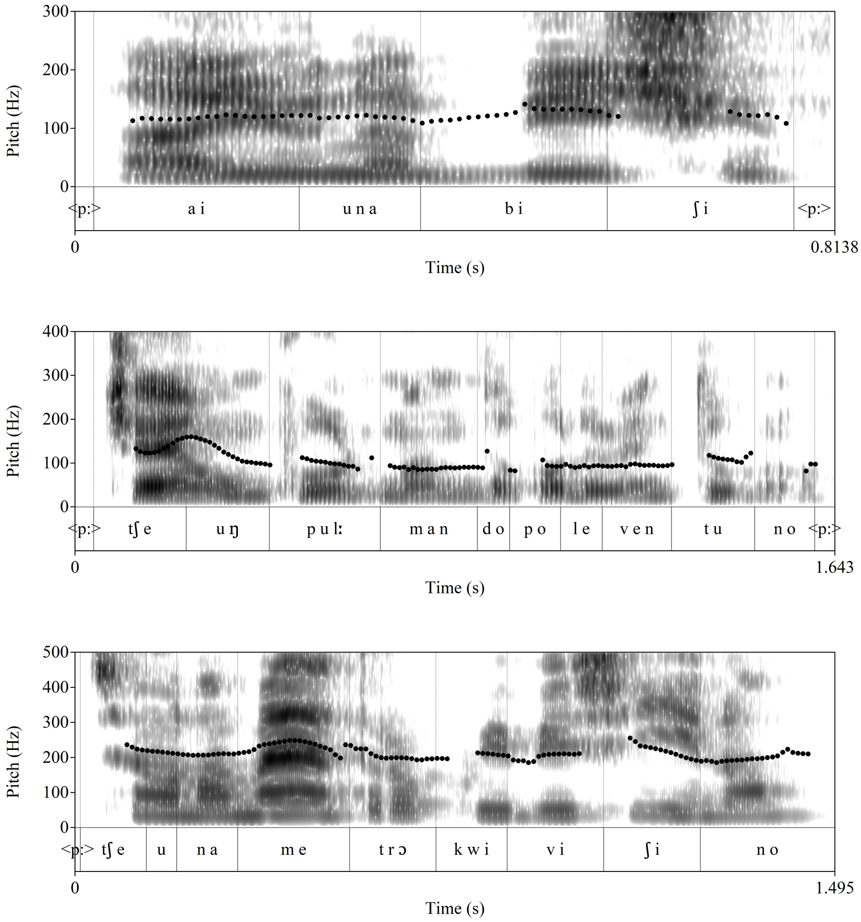


Figure 4.10: PPQs pitch contour in E(-)_E(-) situations: i) Hai una bici? (En. *Do you have a bike?*); ii) C'è un pullman dopo le 21? (En. *Is there a bus after 9pm?*); iii) C'è una metro qui vicino? (En. *Is there a metro station in nearby?)*

as the pitch accent position is concerned, in the first and in the last example the pitch accent is placed at the last lexical item, whereas in the second one on the first lexical item. Interestingly, it was also noticed that, whereas with neutral evidence an early accent was usually used, with a conflicting evidence a late accent was preferred.

RPQ RPQs in $B(\neg p)$_$E(p)$ situations, where the adverb *really* occurs with an epistemic function, differently from the E(-)_$E(p)$ scenario, the boundary tone is generally high, and the pitch accent is placed at an early position, around the adverb *really*, as shown in the examples *Davvero c'è un posto per il campeggio?* (En. *Is there really a camping place?*), and *Davvero hai la ruota di scorta?* (En. *Do you really have a spare wheel?*) in Figure 4.11. In fact, contrary to what shown in [56], where the accent is found on the finite verb, here the adverb has a higher impact, since it is not treated as a separate item with respect to the following PPQ.

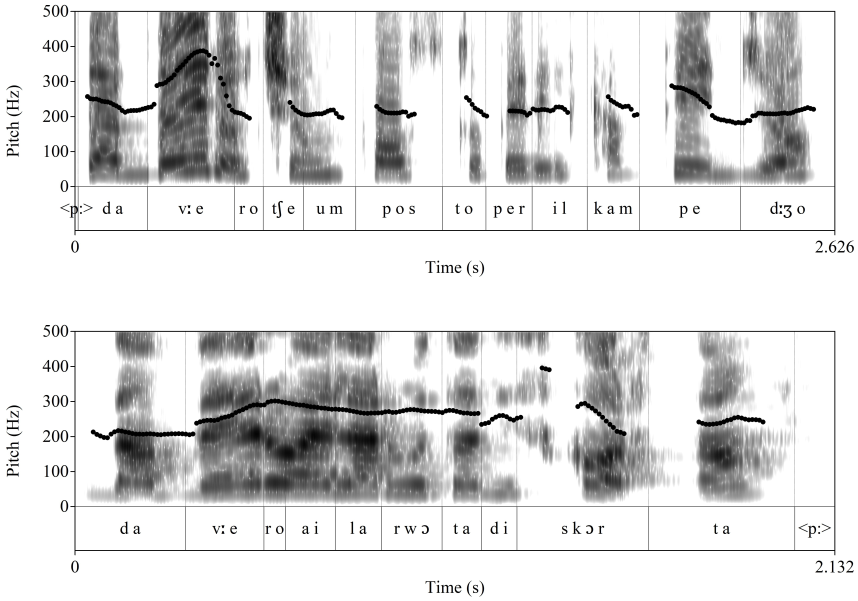


Figure 4.11: RPQs pitch contour in $B(\neg p)\_E(p)$ situations: i) Davvero c'è un posto per il campeggio? (En. *Is there really a camping place?*); ii) Davvero hai la ruota di scorta? (En. *Do you really have a spare wheel?*)

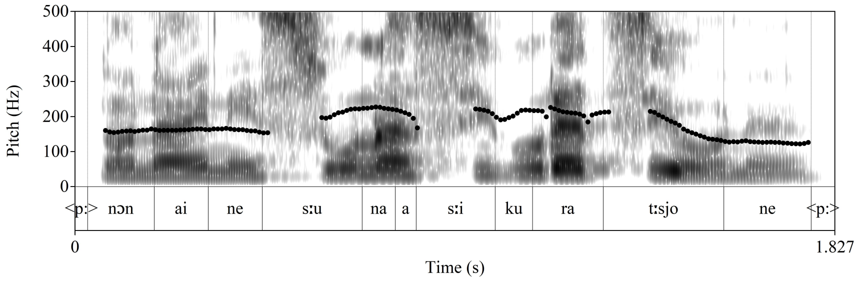


Figure 4.12: LNPQ pitch contour in $E(\text{-})\_E(\neg p)$ situations: i) Non hai nessuna assicurazione? (En. *Don't you have any insurance?*)

LNPQ LNPQs resulted to be a relatively stable class, whose function is linked to the conflict arising from a neutral bias and negative evidence. As its occurrence, its intonational patterns are also stable. In Figure 4.12, the utterance *Non hai nessuna assicurazione?* (En. *Don't you have any insurance?*) is shown. Here, the boundary tone is low, and the pitch accent has a late placement. In fact, the pitch accent is generally positioned around the negative polarity item *nessuno* which represents the focus used to express disaffiliation.

HNPQ HNPQs are frequently used in $B(p)\_E(\text{-})$ and $B(p)\_E(\neg p)$ situations, with a higher frequency for the former situation. In Figure 4.13, both pragmatic functions are displayed. For the former function, the utterance, *Non si vedono le stelle* (En. *Aren't the stars visible?*) is produced with a high

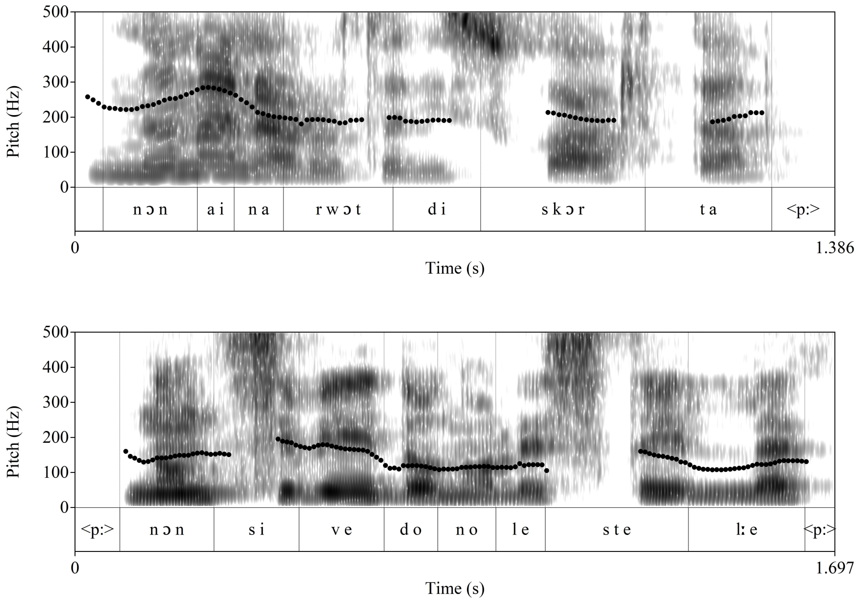


Figure 4.13: HNPQs pitch contour in $B(p)\_E(\text{-})$ and $B(p)\_E(\neg p)$ situations: i) Non hai una ruota di scorta? (En. *Don't you have a spear wheel?*); ii) Non si vedono le stelle? (En. *Aren't the stars visible?*)

boundary tone, and a pitch accent placed on the finite verb. Here, the intonation rises starting from the negation *non*, reaches its peak on the finite verb, and falls before rising again at the end of the question. For the latter function, the utterance *Non hai una ruota di scorta* (En. *Don't you have a spare wheel?*) is shown. Here, the boundary tone is high and the pitch accent has an early placement, as in the previous example. A late pitch accent position is also possible, but when this case occurs the intonation starts rising on the negation adverb and falling on the noun phrase rather than on the final verb.

HNPQ_P HNPQ_Ps whose appropriateness is confirmed for the $B(p)\_E(\neg p)$ situations, are also, in a minor amount, used in $B(p)\_E(\text{-})$ conflicts. Here, despite the low amount of data, a consistent difference can be noticed. For the $B(p)\_E(\text{-})$ situation, the question Non c'era una stampante per tirocinanti? (En. *Wasn't there a printer for trainees?*) is analysed. Here, the boundary tone is high, and the pitch accent has a late position. The same question uttered in a $B(p)\_E(\neg p)$ situation has different characteristics: the boundary tone is here low, and the pitch accent has an early position on the finite verb; here, the intonation starts rising from the negation and falls after the pitch accent. The same pattern is found in another HNPQ_P in the same pragmatic condition, as shown in Figure 4.14. This difference in the pitch accent placement and boundary tone can be traced back to the distinction between inner and outer readings. This distinction was also investigated in [56],

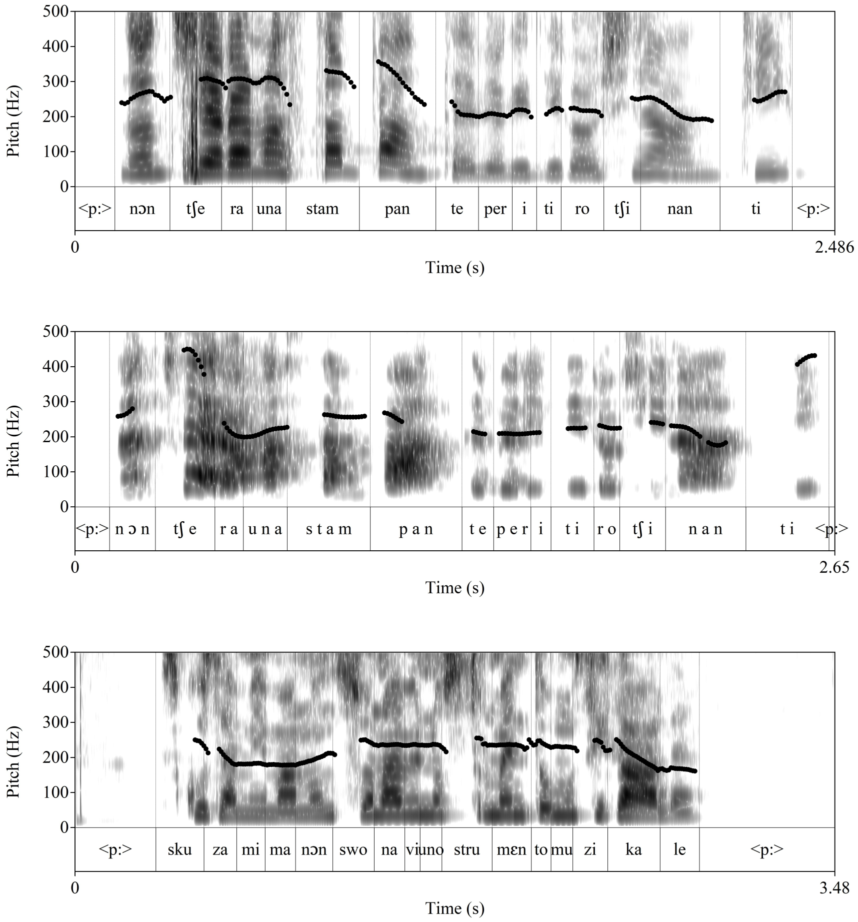


Figure 4.14: HNPQ_Ps pitch contour in $B(p)$_E(-) (first graph) and $B(p)$_$E(\neg p)$ (second and third graphs) situations: i) Non c'era una stampante per tirocinanti? (En. *Wasn't there a printer for trainees?*; ii) Non c'era una stampante per tirocinanti? (En. *Wasn't there a printer for trainees?*; iii) Scusa ma non suonavi uno strumento musicale? (En. *Sorry but didn't you used to play a musical instrumenti?*)

where a rising and a falling pattern were found out for HNPQs. Nonetheless, no further explanations were explored to attribute one pattern to a specific reading. Since the first pattern was found in $B(p)$_E(-) scenarios, it can be hypothesised that this pattern corresponds to the one used for outer readings, whereas the other for inner readings. Nevertheless, more data and targeted experiments are needed to further explore this difference.

The data analysed, therefore, proved that the accent position can have particular importance in the distinction of diverse bias-evidence conflicts. Other authors also described how relevant is to consider the pitch accent placement in polar questions, while analysing the type of bias they express. For instance, in [7], it was demonstrated that the position of the pitch accent on a minimiser or on the finite verb can be interpreted as an expression of a negative bias in polar questions. This was also the case of RPQs, as explained in [56] as well.

On the other hand, the higher impact of pitch accents rather than of boundary tones could depend on the heterogeneity of the sample collected. In fact, although the speakers were all belonging to the same geographical area (Campania, southern Italy), it is important to highlight that in this area, because of its history, different varieties can be microscopically found [8]. For instance, as far as the Salerno variety is concerned, a final rise is allowed in polar questions, whereas in Neapolitan a rise-fall is usually adopted [113]. This shows that among Campanian varieties boundary tones could not be reliable metrics for finding pragmatic tendencies, whereas pitch accents placements seems to be more stable across regional speech types.

4.4 Discussion

This experiment was aimed at testing whether specific forms of polar questions were perceived as more appropriate in specific pragmatic scenarios. The experiment was built upon the one carried out in [56], where scenarios representing different bias-evidence combinations were presented to participants who had to choose the most appropriate question among the ones suggested. In this study, the experiment was, conversely, subdivided in three tasks: the first one (FP) left the participants free to pronounce whatever form they considered to be appropriate to express that particular pragmatic function; the second one (GP) provided the participants with a set of different forms for which they had to give a score of appropriateness; the third one (SS) provided, as the previous one, the participants with a set of synthesised forms to be given a score. In general, the combination of the three tasks of this experiment resulted in the confirmation of the tendencies reported in [56]. Therefore, the **H1**, concerned with proving whether specific forms of PQ were typically used in particular pragmatic scenarios in Italian as well as in German and English, was confirmed. Nevertheless, differently from domaneschi2017bias, the differences resulted to be less sharp, as different forms have similar scores in similar scenarios. This result depends on the annotation protocol which allowed the subjects to express themselves in greater detail, enabling to capture different combinations of pragmatic function and syntactic structure. Specifically, the study shows a clear tendency for preferring HNPQs in the past tense when a positive bias clashes with a negative contextual evidence. Interestingly, although the PPQ is generally the preferred form in the majority of the situations for its versatility, in $B(p)\_E(\neg p)$ scenarios the percentage of scores is lower than in the others. This result leads to the preliminary conclusion that in such situations the adoption of a NPQ better suits the pragmatic needs, increasing the communication efficiency (**H2**). This result is particularly important when considering application scenarios where common ground inconsistency can occur and lead to understanding problems. This is the case of human-machine interaction, for which the adoption of the appropriate form of question can better highlight the nature of the conflict in order to recover it. Further investigation will be conducted in this direction. Specifically, we will investigate whether the use of such a form could also bring to better common ground inconsistencies recovery in human-machine interaction. For this reason, the experiment described in Chapter 5 was set up. The main hypothesis leading the experiment states that the use of the most appropriate polar question when facing common ground inconsistencies, i.e., $B(p)$ versus $E(\neg p)$, can bring to a better understanding of the problem and to its efficient recovery.

Chapter 5

Reporting Common Ground Conflicts in Human-Machine Interaction

The experiment illustrated in Chapter 4 pointed out the importance speakers give to the syntactic form with respect to the pragmatic needs, specifically as far as polar questions are concerned, here under investigation. The results showed that the use of high negation polar questions better suits the pragmatic need of referring to a specific type of conflict between an original bias and an opposing contextual evidence. Namely, the conflict is between a strong presupposition of the speaker and a piece of information stored in the *Personal Common Ground* in a previous step of the interaction clashing with a contextual evidence given by the interlocutor. Given this, the research question RQ3 (see the Introduction) driving the second experiment is about testing whether the relationship between the syntactic and the pragmatic level is only caused by naturalness principles, or whether it also has practical consequences. In fact, language patterns exist because they serve peculiar aims. In other words, speakers make use of specific signals to achieve some communicative goals and consequently produce practical consequences. The same principles can, therefore, be applied when modelling human-machine dialogues. As a consequence, the importance of the results collected from the previous experiment is to be considered in the field of cognitive pragmatics and its computational applications. In fact, the study of the mental states of the interlocutors involved in a conversational exchange is important for practical applications, such as in dialogue systems. These systems can, indeed, rely on the understanding and representation of mental states, either of their own or of the interlocutor, to encode and decode the correct relations between the language usage and contextual characteristics, among which intentions and presuppositions are listed. For this reason, even an apparently marginal difference, like the use of a negated form against its positive one, can express a specific speaker's stance and have a strong impact on the conversation efficiency, as far as, for example, robustness is concerned. Starting from these motivations, another experiment was planned. Contrary to the previous experiment, whose aim was to understand which form was naturally appropriate according to the type of conflict, this second study was, namely, aimed at putting participants not in a narrated conflicting situation, but in a real conflicting context, simulating what could happen in human-machine interaction and how the forms previously considered appropriate could improve the interaction quality. In this way, the study of the behaviour caused by the use of one form or another could be studied in a real context of use. For the setup of the experiment, specific syntactic forms from the previous experiment were selected and applied in the present experiment dealing with a simulated human-machine interaction. The forms which were selected are the PPQs, which were the most frequent forms in general, and the HNPQs, which were rated to be the most appropriate ones, as far as the $B(p)\_E(\neg p)$ scenario was concerned. This contrast corresponds, indeed, to the inconsistencies which *Common Ground* CRs question about, focus of this study. In the following sections, the research hypothesis and the setup for carrying out the experiment are presented, before describing the results and their human-machine interaction application[1].

5.1 Conflict-related Correcting Feedback in Conversational Agents

Different scholars highlighted the urge of including corrective dialogues in their systems to improve the communication process. This need resulted from the users' need to interact with an agent capable of cooperating to the communicative actions. Human interlocutors always contribute with questions, answers, and feedback [16]. A corrective dialogue is a particular type of sub-dialogue which occurs when: i) the user notices an error in the system and corrects it; ii) the user changes their mind; iii) the user's beliefs are in contradiction with the system's beliefs and expectations. In the first two cases, the corrective dialogue is initiated by the user, whereas, in the last case, it is initiated by the system [22]. One example of corrective dialogue in human-machine interaction is the one presented in [16]. The authors focused on a particular communicative problem related to conceptual discrepancies between a computer system and its user. Starting from the assumption that both the system and its user have a mental representation of a domain, the mental representation of the

[1] The content of this chapter is also described in [52]

system, also referred to as ontology, contains conceptualisations that are made explicit in a formal language. Although they are usually incomplete and inaccurate, this information can be used to trace the system's reasoning about the concepts, items, and their properties. This representation also allows the detection of conceptual discrepancies, for example when the system observes that the user applies an incorrect action to a particular object. The authors also stated that, although feedback is now used in such systems, there is still no accurate *mathematical theory* for natural communicative behaviours and their computational model to human-machine interaction, especially as far as conceptual discrepancies are concerned. What is still missing is, therefore, a reference model guiding the adoption of a specific type, content, and form of the feedback that has to be generated in a particular situation [16]. For example, the choice could depend on different reasons: i) the domain knowledge in both the system and its user, more specifically, the system's knowledge about the user's conceptualisation; ii) the role played by the system in the interaction (i.e. whether the system is a the expert or not), a parameter which affect the willingness to adjust the ontology.

In this work, a type of corrective dialogue is investigated, in which the system has a non-expert role and adjusts its grounded knowledge when conceptual discrepancies occur because of an inaccuracy, which causes an inconsistency, in the sequence of actions uttered by the user. The type and form of feedback are here investigated, not only as far as the appropriateness derived from the experiment in Chapter 4 is concerned, but also for the practical effects that act on the interaction itself.

5.1.1 Research Hypothesis

As previously described (Chapter 4), negative polar questions, especially in the past tense, can be used to express a positive bias against a negative evidence, contrary to PPQs which were considered inadequate in this contrasting condition. In the present study, both polarities were tested against a general error signal, in order to prove that the adoption of the negated form can actually improve the interaction efficiency while conversing with dialogue systems as well. Nowadays, in both commercial and academic dialogue systems, general error messages are usually used for their versatility. However, the research question that motivates this experiment lies in the understanding of how explicative these error messages actually are for human users in order to get which understanding problem occurred and how to repair it, or, conversely, which polar question form can substitute it and better suit the goal. Specifically, it is intended to verify how the positive or negative form of the question is indicative of the nature of this particular contrast: *is the contrast between* *the speaker's belief and the contextual evidence better communicated by using a specific question form (i.e. positive or negative polar questions)*? The hypothesis, based on the previous experiment, states that, when using a negative polar question, the positive bias of the speaker, or their mental state, toward a presupposition part of its *Personal Common Ground* is better displayed and easier to interpret by the interlocutor, who can try to solve the inconsistency or explain, or even teach, the reason of its instantiation.

As anticipated in Chapter 3, from an interaction design point of view, this has a potential effect on the principle of robustness, which requires: i) observability, i.e. the extent to which the user can evaluate the internal state of the system given the representation provided by the user interface; ii) recoverability, i.e. the extent to which the user can achieve the intended purpose after recognising an error in the previous interaction; iii) task compliance, i.e. the extent to which the services provided by the system support all tasks that the user may wish to perform, in the way they wish to perform them; iv) reactivity, the measurement of the communication speed between the user and the system. In particular, what can actually take advantage of the correct use of clarification requests in terms of consistency between the form used and the type of problem related to the Common Ground to be reported are the *observability* and the *recoverability* measures.

In the next sub-sections, the experiment is described along with the result analysis, before explaining the practical application of the findings in Chapter 6.

5.1.2 Experimental setup

The second experiment consisted in a series of slides $S = \{s_1 ... s_n\}$, for each of which participants were asked to elaborate spoken commands. Each s_i was designed to represent an action to be performed to complete a recipe. To simulate the conflicting situation, for whose resolution a consistency recovery strategy had to be employed, an inconsistent action s_x was inserted in S. The inconsistency emerges when the pre-conditions of a later action are not verified because of s_x. The conflicting inconsistency, representing a positive bias versus negative evidence contrast, was determined by the opposition of some aspects of s_x and some aspects of the consecutive s_n. The main goal of the experiment was to check if a specific error message, shown in s_q, was useful to signal to the user the existence of a conflict arising from s_x and its details in a succinct, natural way. Furthermore, to reduce the possibility that the s_x was in the subject's short term memory, once the conflict arised, making it easier to detect, it was positioned at a minimum distance of five s_n from s_q.

In other words

$$q - x \geqslant 5 \tag{5.1}$$

The experiment was divided in three parts: in the first one, the control group was used to check the behaviour participants adopted when the inconsistency was shown through a general error message (i.e. *This action is not possible because it clashes with a previous one*); the second one was tested the behaviour adopted, in case a positive polar question was used; the third one was aimed at testing how participants behaved when the error was expressed through a high negation polar question. Specifically, it was hypothesised that in case of conflicts between user beliefs (positive bias) and an opposing contextual observation (negative evidence), the use of a negative polar question would increase the observability degree of the internal state of the dialogue system and would decrease the required time for the recovery of the inconsistency problem.

The application domain of this experiment and of the resulting dialogue system (Chapter 6) is the cooking domain. This domain was chosen for three important reasons: i) the familiarity with this domain is presumably high among speakers, being part of the everyday life; ii) similarly to the map-task, this domain could be applied in a deliberative dialogue, characterised by the process by which two or more agents reach a consensus on a course of action - a type of dialogue important for *User Guided Task Applications* (Chapter 3); iii) contrary to the traditional map-task, the number of different actions is higher, making the tasks more varied and slightly more articulated; moreover, single actions, although atomic, are often linked to each other, in the sense that an action can affect a consequent one. The combination of these three reasons made it possible to create situations in which inconsistencies could be inserted.

For the preparation of the experiment, ten different recipes were taken from the Italian recipes' website *Giallo Zafferano*[2]. See table 5.1 for the list of recipes. For each recipe, single steps, or intents, were identified and semantically annotated in XML files. For the semantic annotation, the frame semantics methodology was adopted [115, 59]. Semantic frames are defined as conceptual structures evoked by action words in the mind of a speaker. Each frame can be linguistically expressed when the action words are syntactically combined with phrases bearing specific semantic and syntactic roles, i.e., frame elements. Frames were taken from the lexical database FrameNet[3] whose word senses descriptions are based on the framework of frame semantics [9]. Below, the annotation procedure is shown

[2] https://www.giallozafferano.it/

[3] https://framenet.icsi.berkeley.edu/

```
<intent frame='Apply_heat'>
  <frame>
    <fe name='Food' type='ingredient' property='cooked'>latte</fe>
    <fe name='Container' type='cookingTool' property='none'>pentolino</fe>
  </frame>
</intent>
```

Here, for the intent *boil the milk* (it. *lascia bollire il latte*), the corresponding frame is [Apply_heat]. In FrameNet, this frame is described as follows

Frame 1. ***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. A less semantically prominent Food or Cook is marked Co-participant.*

In this definition, the core frame elements, representing the semantic roles, necessary to express particular meanings of actions, are also mentioned (underlined in the frame definition), such as Food, Temperature_setting, Duration, Heating_instrument, Medium, and Cook. Although most of the frame elements are described in terms of semantic types, i.e. the specific category a frame element can belong to (animate vs. inanimate, time, location, manner, sentient, etc.), some of them are missing or are too generic. This is because FrameNet is intended to be a both human- and machine-readable open-domain lexical database for domain applications. Further specifications were, therefore, needed. For this reason, the argument *type* was added, as shown in the aforementioned annotation example. For instance, in the cooking domain, Food, Patient, New_member, Parts, in specific frames, are classified as *Ingredient*, some frame elements like Grinder are tools, and so on. This is important to define specific slots which can be filled by pre-defined object classes. Each frame element is also described as far as the properties regarding their state after the processing of the intent itself. This is what we call *post-condition*. In the example, the frame element Food has the property *cooked* when used within the frame [Apply_heat]. This means that, from that point in time onwards, that ingredient can only be used when a next action to process accepts the *pre-condition* of being cooked. For the representation and application of pre- and post-condition, details are given in section 6.3. In total, beside the aforementioned frame Apply_heat, the following eleven frames were used to annotate the recipes:

Frame 2. ***Cause_to_amalgamate.*** *An Agent or Cause makes a New_member part of Group. The Group may be represented by an individual Existing_* *member if it implies the existence of a set of members.*

Frame 3. ***Cause_to_be_included.*** *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 that are easily discernable or separable in the Whole.*

Frame 4. ***Cutting.*** *An Agent cuts a Item into Pieces using an Instrument (which may or may not be expressed).*

Frame 5. ***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 6. ***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 7. ***Manipulation.*** *The words in this frame describe the manipulation of an Entity by an Agent. Generally, this implies that the Entity is not deeply or permanently physically affected, nor is it overall moved from one place to another.*

Frame 8. ***Placing.*** *Generally without overall (translational) motion, 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/Cause at the time of its arrival at the Goal.*

Frame 9. ***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 10. ***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. Some of these words indicate that the Configuration is an undesirable alteration of the norm, and in such cases the lexical unit is marked with the semantic type Negative_judgement. This frame does not include senses that specifically indicate causing harm to a living being.*

Frame 11. ***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.*

	R1	**R2**
S1	Tiramisù	Polpettine
S2	Piadina	Cestini ripieni
S3	Cestini ripieni	Carbonara
S4	Besciamella	Piadina
S5	Patate al forno	Besciamella
S6	Crocchette di patate	Pizzette rosse
S7	Carbonara	Pancakes
S8	Pancakes	Patate al forno
S9	Pizzette rosse	Tiramisù
S10	Polpettine	Crocchette di patate
S11	Piadina	Pancakes
S12	Pancakes	Piadina

Table 5.1: Recipes' distribution for each experimental session

Frame 12. ***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.*

As for the map-task, for the assignments arranged in this experiment, visual stimuli, in the form of slides $S = \{s_1, \ldots, s_n\}$, were used. In this way, a potential influence of linguistic material over participants' spoken commands was avoided. Moreover, to make the task cognitively not heavy for the users, the slides' structure was kept coherent, i.e., the same stimulus for the same action must be used. Hence, each intent was represented using a fixed structure: given the set of actions A (i.e. *mix*, *boil*, etc.), and given $a \in A$, the abstract action a was represented on the left side of the slide through a dynamic image[4]; given the set of ingredients G and the set of cooking tools T, the set of parameters $P = p_n$ corresponded to $P = G \cup T$, represented on the right side of the slide through static images (Figure 5.1 shows an example). The meaning of an intent I is, therefore, completed when a is combined with one or more parameters p . In other words,

$$I = (a, [p_1, \ldots, p_n]) \tag{5.2}$$

I defines a specific domain's action which is the result of the combination of a with one or more elements of P. More precisely, different $p \cup P$ can be of the type *ingredient* or $cookingTool$, where the first one is a list of one or more items (note that a set of items can be substituted by holonyms, such as *mixture*), compulsory for the construction of I. The compulsory property of this list is, nevertheless, not linguistic, but semantic, as aforementioned ingredients or mixtures can be left out of the utterance in some contexts. The intelligibility of the structure of the experiment was confirmed after the first phase of

[4] Gifs were generated from video recipes taken from *Giallo Zafferano*

Recipe	**Code**	**Conflict Type**
Besciamella	R01	Quantity
Carbonara	R02	Ingredient
Cestini ripieni	R03	Ingredient
Crocchette di patate	R04	Quantity
Pancakes	R05	Quantity
Patate al forno	R06	Quantity
Piadina	R07	Quantity
Polpettine	R08	Quantity
Tiramisù	R09	Ingredient
Pizzette rosse	R10	Ingredient

Table 5.2: Recipes tested within the experiment with their conflict type description

	Control		**PPQ**		**NPQ**	
	R1	**R2**	**R1**	**R2**	**R1**	**R2**
S1	✓	✓	✓			✓
S2			✓			
S3			✓	✓	✓	
S4	✓	✓	✓		✓	
S5			✓	✓	✓	
S6			✓		✓	✓
S7	✓				✓	
S8	✓	✓			✓	✓
S9				✓	✓	✓
S10				✓	✓	
S11	✓	✓			✓	✓
S12			✓		✓	✓

Table 5.3: Distribution of conflicts found by each subject

Recipe	**Code**	**Conflict Found**
Besciamella	R01	66,67%
Carbonara	R02	50%
Cestini ripieni	R03	33,33%
Crocchette di patate	R04	50%
Pancakes	R05	75%
Patate al forno	R06	66,67%
Piadina	R07	62,5%
Polpettine	R08	50%
Tiramisù	R09	66,67%
Pizzette rosse	R10	33,33%

Table 5.4: Percentage of conflicts found per recipe

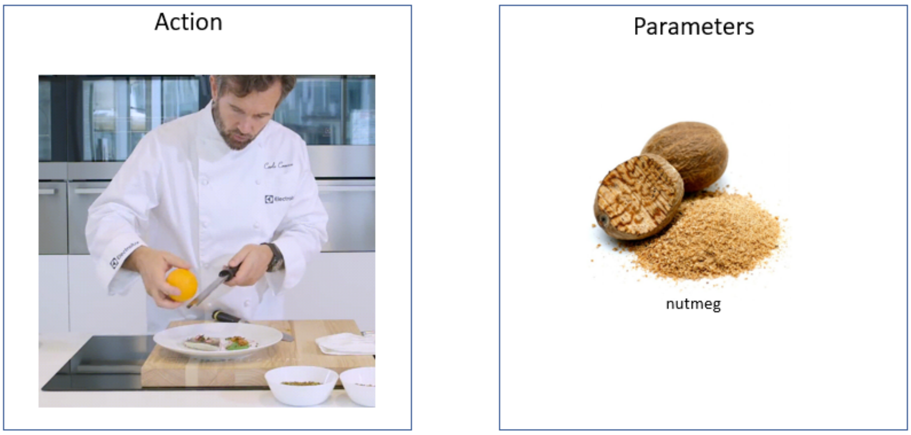


Figure 5.1: A slide from the experiment; $[I = a\{grate\} + p\{the\_nutmeg\}]$

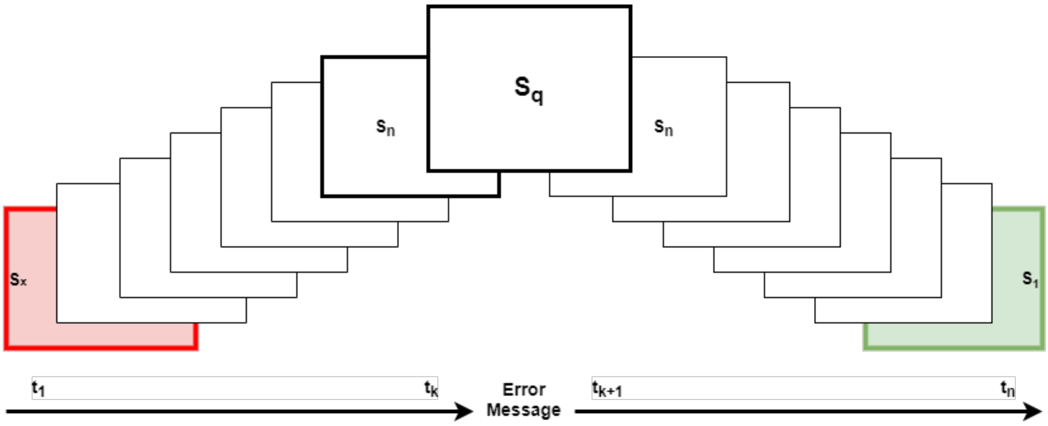


Figure 5.2: Experiment structure

the study, during which participants got familiar with its structure through a training recipe. After the training phase of the experiment, participants did not encounter any problem with the formulation of the commands, confirming that the structure of the experiment was coherent.

For each recipe, a conflicting slide s_x was introduced to semantically link the prompt signalling the inconsistency to it. The conflicts were of two different types: quantity-related or ingredient-related. Quantity-related conflicts refer to the situation where an ingredient is used without specifying the quantity; in fact, when this specification is missing, the interlocutor presupposes that after the action is processed the ingredient is no longer available, except for spices which are considered to be always available. Furthermore, spices were treated differently also because they could not be interpreted metonymically as reference for mixtures, differently from other ingredients such as *flour*. On the other hand, ingredient-related conflicts refer to ingredients which have been used as part of a preceding action instead of the correct ingredient. This makes them no longer available, although they would have been if the correct ingredient was used before. Other possible conflicts, which were not considered in this experiment, can refer to the state of the ingredient; in fact, if an ingredient is liquid, it can't be cut, or if it is ground, it can't be grated, and so on. In table 5.2, the conflict types for each recipe are summarised.

Once the prompt message s_q was shown, participants could go back in the recipe in order to look for the conflict. The experiment, which made use of slides, was constructed in a way that, once the subject requested to go back after the prompt, the experimenter went instead forth, where the previous slides where presented backwards, as shown in Figure 5.2. Here, the conflicting slide was substituted with the correct one s_1. In this way, the identification of the conflict and the speaker's self-correction could be guided. Furthermore, it is important to remember that the participants thought that the goal of the experiment was to decode the slides and elaborate spoken commands, whereas the hidden aim was to test the speakers' behaviour in a real conflicting situation where the conflict was instead narrated in the experiment presented in Chapter 4. In fact, the instructions given to participants were the followings (here translated from Italian)

> In this experiment, we ask you to look at a series of slides describing a recipe and tell the experimenter what to do to make it. Each slide is made up of actions and parameters: actions generally describe what to do, while parameters show which objects are involved in the action. In case of problems, you are free to move back and forth in the recipe as you like by asking the experimenter in which direction to move the slides.
>
> Take your time to think about what to say and give your instructions when you are sure they are correct.
>
> We first start with a training recipe, in which you are free to ask any questions you want to the experimenter. Once the test starts, the experimenter will no longer be able to answer you, except for carrying out your instructions until the end of the experiment.

This strategy was important to artificially re-create a natural interaction in which participants could be unaware of the real aim of the task, concerned with finding and solving a conflict.

As mentioned before, three different experimental conditions with the same recipes and conflicts were carried out, as follows

- Control group: the first experimental condition consisted of a combination of two recipes as shown in table 5.1; this condition was used both as validation for the experimental setup, in order to understand if the slides and the task were understandable for the participants, and as analysis of the general error message which was used to signalise the conflict (i.e. *This action is not possible because it clashes with a previous one*). The resulting collected values were, therefore, used as a term of comparison for statistical analysis.

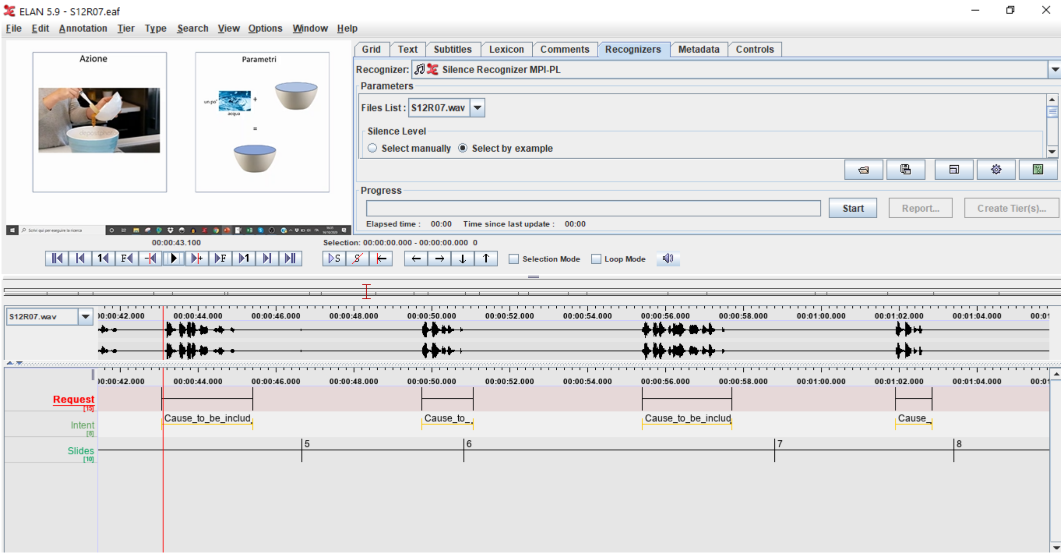
- PPQ group: the second experimental condition differed from the previous one just for the error message, where a positive polar question was instead used (i.e, *Did I have to add the flour to the container?*). The use of the most frequent polar question form was useful to test its appropriateness in bias-evidence conflicts in simulated human-machine interactions.
- NPQ group: the third experimental condition, similarly as the previous one, made use of a negative polar question, and more specifically a high negation polar question in the past tense (i.e, *Didn't I have to add the flour to the container?*), whose appropriateness in the positive bias versus negative evidence scenarios was confirmed in the experiment described in Chapter 4. This condition was hypothesised to bring to better results in terms of observability and recoverability.

For each experimental condition, 12 participants were employed, for a total of 36 testers. For the control group, the average age was of 25,5, with an average self-evaluation of their cooking skills equal to 2,33 (on a scale from 1 to 5). For the PPQ group, the average age was of 27,25 with 2,67 points of cooking skills. Finally, for the NPQ group, the participants were on average 26,08 years old and their cooking skills were around 2,92 points. The experiment was carried out online, through Skype[5]. The screen and the voice were recorded during the session. The resulting videos were used to analyse the participants' behaviour in order to compute the results, as far as observability and recoverability were concerned. Further details concerning the video annotation and the resulting analysis are given in the next section (Section 5.1.2.1).

5.1.2.1 Analysis and Results

In this section, the collected data analysis and the consequent results are described. First of all, each video was annotated via ELAN [178], as in figure 5.3. As a first step, for each video, the silence recognizer integrated in ELAN was used to automatically annotate the segments containing speech signals. Each segment was reported in the tier called *Request*. Each request was then annotated with the name of the corresponding intent (i.e. in the tier *Intent*), which represented a specific frame as described in Section 5.1.2. In the tier *Slides*, on the other hand, the sequential number of the slide is reported, following a

[5] This was due to COVID-19 restrictions


ELAN 5.9 - S12R07.eaf
File Edit Annotation Tier Type Search View Options Window Help
Azione
Parametri
Grid Text Subtitles Lexicon Comments Recognizers Metadata Controls
Parameters
Silence Level
Progress
Start
Report...
Create Tier(s)...
Elapsed time : 00:00 Time since last update : 00:00
00:00:43.100
Selection: 00:00:00.000 - 00:00:00.000 0
Selection Mode
Loop Mode
Request
Intent
Slides
Cause_to_be_includ
Cause_to_
Cause_to_be_includ
Cause_
5
6
7
8

Recognizer: Silence Recognizer MPI-PL

Files List: S12R07.wav

○ Select manually ● Select by example

Figure 5.3: Example Annotation in ELAN

semi-automatic procedure: PySceneDetect[6] was used to automatically recognise the different slides based on differences between scenes; results were then manually corrected. The scene information was useful to compute the time participants spent on each slide and to understand how they moved across the presentation. In other words, this information was useful to answer the following questions: How much time did they need to find the conflict? How many times had they have to go back and forth before finding the conflict?

First of all, before analysing the results related to the time spent on each slide after the occurrence of the conflict and, consequently, the time spent finding the conflict itself, the number of conflicts found was considered. Using the binomial test, a non-parametric test for binary variables [170], the deviations from a reference distribution of observations was computed. The test showed that, when using NPQs, the conflict was found more frequently in a statistically significant way ($p = 0.005$) when compared to the Control group, whereas, when using PPQs, the difference resulted not to be statistically significant ($p = 0.4$), as shown in Table 5.5. In table 5.3, the distribution of the conflicts found related to recipes and subjects are shown. Three aspects are clear from this table: i) in the NPQ condition, conflicts were found more frequently than in the other two experimental conditions, more specifically they were found 66.67% of the times, as also shown in table 5.6; ii) there is no correlation between recipe and conflict found, as also shown in table 5.4 (note that *Piadina* and *Pancakes* were presented four times instead of two differently from the other recipes); iii) the conflicts tended to be found in the first recipe, suggesting that, on the one hand, participants were probably tired after the

[6] https://pyscenedetect.readthedocs.io/en/latest/

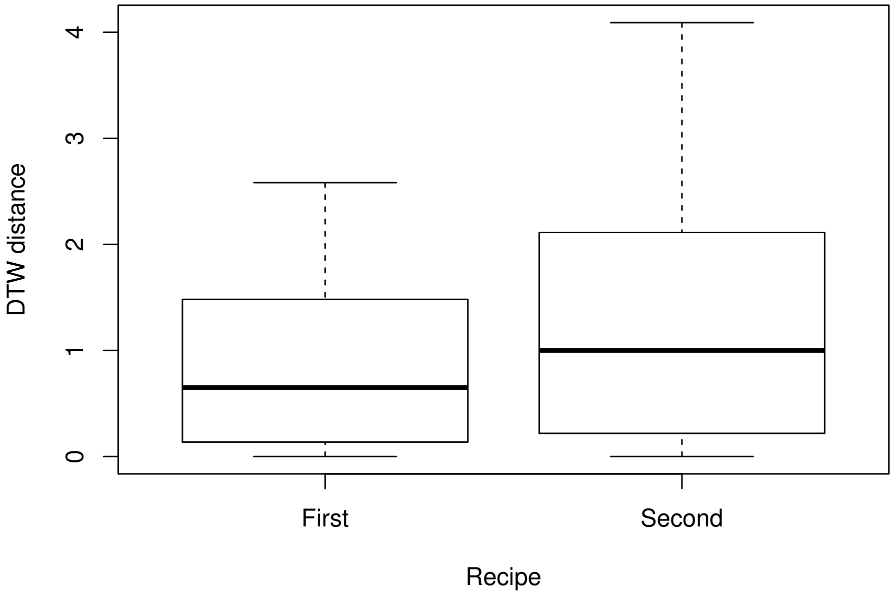
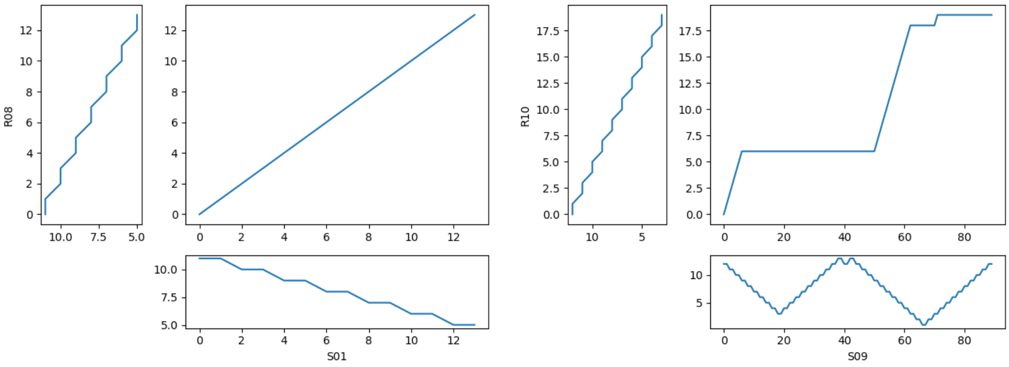


Figure 5.4: Box Plot showing differences between the first and the second recipes

training and after the first recipe, as shown in the box plots in figure 5.4, where for the second recipes they were slightly slower, although with no statistically significant difference ($p = 0.2268$; this value resulted from Wilcoxon rank sum test with continuity correction).

As far as recoverability is concerned, the difference between the sequence of moves the users made to reach the conflicting slide and the optimal sequence they would have followed, if aware of the error after being signalised by the prompt, was computed using the Dynamic Time Warping [108], through the Python dynamic time warping library and R [66]. The Dynamic Time Warp-



(a) Direct pattern with distance 0

(b) Sinusoidal pattern with distance 1.93

Figure 5.5: Example Patterns for Dynamic Time Warping Distance

	2R	**1R**
PPQ	0,4	0,26
NPQ	0,005	0,005

Table 5.5: Binomial test results

	R1	**R2**	**Tot**	**Perc**
Control	5	4	9	37,5
PPQ	7	4	11	45,83
NPQ	10	6	16	66,67

Table 5.6: Number of conflicts found in the three experimental setups (per recipe, in total, and in percentage)

	Control	**NPQ**
NPQ	0.089	-
PPQ	0.75	0.39

Table 5.7: Pairwise comparisons using Wilcoxon rank sum test (adjustment method: holm)

ing algorithm is used to find an optimal alignment between time-dependent sequences [108]. An example of a user who goes directly on the conflicting slide is given in in Figure 5.5a, whereas an example of a user not sure of which slide caused the problem is shown in Figure 5.5b. In absolute terms, users spent less time, on average, looking for a solution when the high negation polar question was used, confirming its appropriateness and consequent recoverability effect on the interaction, when compared with a general error message and a PPQ. Nevertheless, this difference resulted not to be statistically significant ($p = 0.089$), as shown in table 5.7. More specifically, since the distributions were not normal, as proved with the Shapiro-Wilcoxon test [147], to test the collected distances, the Kruskal-Wallis test [89] was adopted. This was, indeed, useful to test whether the median ranks of the groups were the same. In Figures 5.6 and 5.7, Dynamic Time Warping distances are shown in order to represent the time spent to find the error in the sequence of actions.

As far as the conflict prompt understanding is concerned, it was also hypothesised that, since the PPQ was generally perceived more as a confirmation request rather than a conflict signalling message, as also suggested by one of the participants during the test, more time was required to understand the prompt as a conflict signal when shown for the first time. To do that, the duration time values spent on the prompt slide were taken into account for each experimental group. On average, for the PPQ prompts, more time was required when compared to the adoption of a NPQ or a general message (35,52 seconds), suggesting that the conflict is not well signalised through a PPQ. For the error message, on the other hand, the Control group required, on average, 22,24 seconds to understand what to do next. Finally, the NPQ was under-

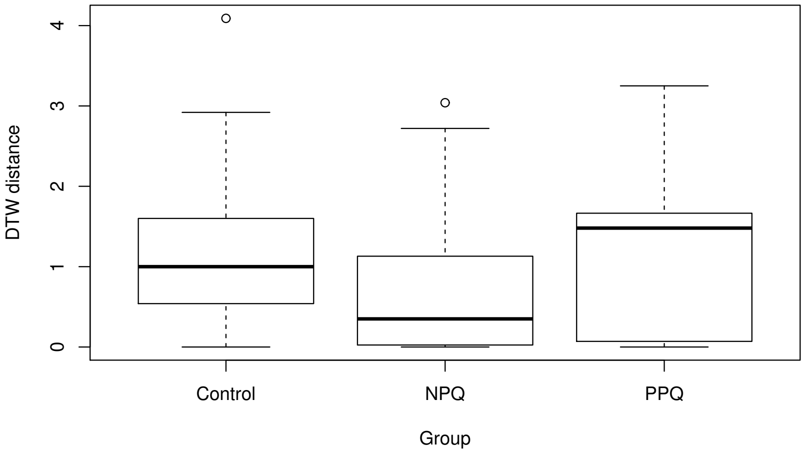


Figure 5.6: Box plots representing distances distribution in the three experimental conditions

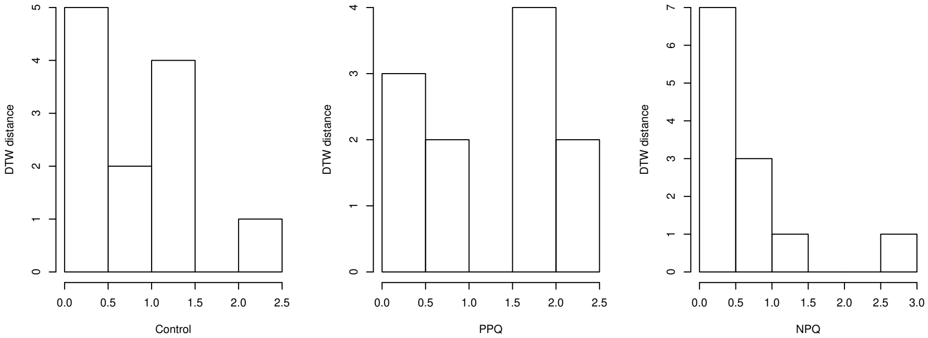


Figure 5.7: Histogram representing distances distribution in the three experimental conditions

stood in 26,44 seconds. All in all, as the presence of a conflict was already suggested in the text of the error message, the Control group needed less time to understand what the aim of the prompt was, although not being that useful to observe and recover the conflict itself, as previously illustrated. Nonetheless, the NPQ was found averagely only 4 seconds later than the error message, and it was more effective for observing and recovering the problem. The PPQ was, instead, interpreted almost around 10 seconds later than the other prompts, suggesting its greater inadequacy. The difference between the duration distributions is, nevertheless, not statistically significant ($p = 0.37$), although the tendency appears to be clear in the box plots in figure 5.8.

All in all, NPQs signalling Common Ground inconsistencies led to an increased observability of the problem found by the agent, as demonstrated by the fact that conflicts were found more frequently when this type of prompt was used. Since they also led to an increased recoverability of the inconsistency,

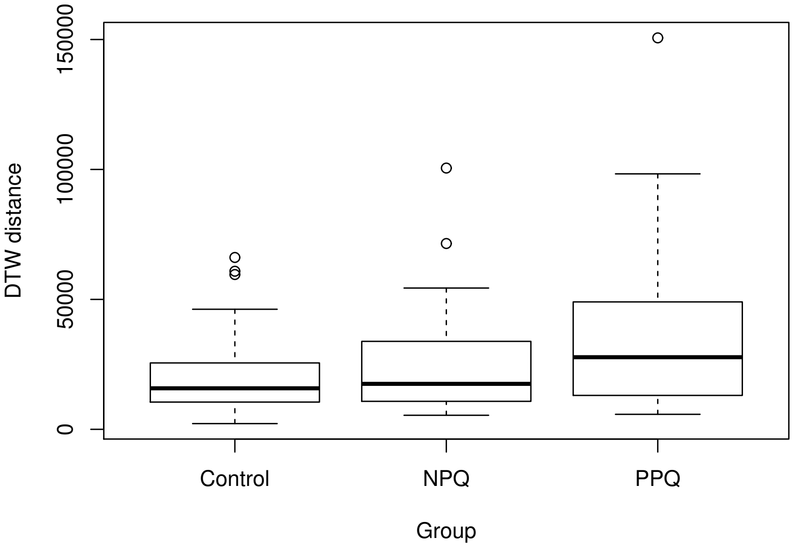


Figure 5.8: Prompt Understanding Differences (milliseconds)

as shown by the inferior time values spent to find the conflict, the adoption of this syntactic form is confirmed to be appropriate in $B(p)\_E(\neg p)$ scenarios in human-machine interactions. Proving the efficiency of such a form in a defined pragmatic situation, like the one tested in this experiment, can be used to good advantage in dialogue systems aimed at learning sequences of actions uttered by a human interlocutor. Such systems are the so-called *User Guided Tasks Applications* (Chapter 3). In case of pre- and post-conditions-based inconsistencies between two uttered actions, the system can use a knowledge representation module, as explained in Chapter 6.3, to recognise the problem and signalise it by using a NPQ. In fact, the type of conflict arising represents a $B(p)\_E(\neg p)$ condition, as a preceding action, part of the Personal Common Ground, becomes a system presupposition, whereas the new uttered action, which is in conflict with an expected pre-condition, represents the negative evidence.

In the next Chapter, the proposal for an argumentation-based dialogue system architecture is presented. This is expected to be able to use Common Ground representation, to highlight possible conflicts in the Common Ground, to signal them in an efficient way, specifically with a clarification request in the form of a HNPQ_P. To do so, a conflict search graph, described and tested in the next Chapter, is used to let the system be aware of the conflict.

Chapter 6

An Argumentation-based Dialogue System Architecture

In this Chapter, a proposal showing how the results obtained in the previous experiments can be modelled in a spoken dialogue system capable of dealing with common ground inconsistencies and signal them via the appropriate question form. Given the results obtained in the previous experiments, providing automatic dialogue systems with such capabilities can lead to improved usability and naturalness. Such a system is, thus, able to detect conflicts and to use argumentation strategies to signal them consistently with previous observations. In the next sections, the system architecture is described. Specifically, attention will be drawn upon the Conflict Search Graph, with insights on its ability to recognise problems and make them explicit via polar questions. System performances in speech and intent recognition are also provided[1].

6.1 System Architecture

The system presented here is intended as one of the possible applications of the framework FANTASIA by Origlia et al. [110], whose architecture is shown in Figure 6.1[2]. FANTASIA's aim is to integrate different modules, such as a graph database, a dialogue manager, a game engine, and a voice synthesis engine for the development of social interactive systems. Integration efforts are, indeed, an important issue to overcome when a research group, for instance, shares the same theoretical framework but needs ad-hoc solutions for different applications. Different approaches typically concentrated on communication layers, to which different actors in an interactive system must subscribe to exchange data. In such approaches, developing low-level code is still necessary to implement the application. Contrary to these approaches, the high-level

[1] The content of this chapter is also described in [49]

[2] Figure 6.1 shows an improved version of the architecture of the one displayed in the reference paper [110]

development languages provided by game engines, but also by other specialised solutions, offer an important chance to simplify the process when directly integrated in a proposed framework, as in FANTASIA.

The application of interest in this work is concerned with natural interaction. Specialised frameworks have dealt with this kind of interaction and focused mainly on virtual human management. In these frameworks, when game engines are adopted, they have usually been used only as rendering modules. However, modern game engines are interesting candidates to host most of the behavioural logic and realisation modules in an integrated solution. In FANTASIA, as shown in Figure 6.1, not only is a powerful game engine such as the Unreal Engine 4 (UE4) [134] adopted to control the virtual environment and the virtual human (in this work, the virtual robot named Bastian) communicating with the human user, but it is also used to integrate language processing pipelines, and Bayesian reasoning by relying on informational data received by domain representations based on external resources. For natural language processing, both speech and intent recognition, Microsoft Azure [46] was adopted. Specifically, the LUIS[3] (Language Understanding) service was used. LUIS is a cloud-based conversational AI service that applies machine learning approaches to natural language text to predict intents, meanings, and retrieve relevant, detailed information. On the other hand, the knowledge base was represented in a graph database developed in Neo4j. Neo4j [175] is an open source graph database manager that has been developed over the last 16 years and applied to a high number of tasks related to data representation. It can be deployed in server mode and queried over a specific port using a standard HTTP or the dedicated Bolt protocol. It can also be embedded in Java applications through dedicated APIs. In Neo4j, nodes and relationships may be assigned *labels* that describe the type of object they are associated with. Neo4j is characterised by high scalability, ease of use and its proprietary query language, namely Cypher. Cypher is designed to be a *declarative* language that highlights patterns' structure using an SQL-inspired *ASCII-art syntax*. The increasing importance of graph databases is also pointed out in the *Gartner Top 10 Trends in Data and Analytics for 2020* where graph analytics and algorithms are considered important to improve AI and ML initiatives[4]. Furthermore, The increasing importance of Neo4j is also demonstrated by the fact that this tool is able to detect conflicts and to use argumentation strategies to signal them consistently with previous observations. This means that such graphs can be employed not only for rule-based reasoning but also for machine learning approaches.

[3] https://www.luis.ai/

[4] https://www.gartner.com/smarterwithgartner/gartner-top-10-trends-in-data-and-analytics-for-2020/ [last consultation on 19th January 2021]

Neo4j allows to combine data coming from different sources under a single, graph-based representation; for instance, sources of information other than textual and Linked-Open Data (LOD) can be integrated in the representation, like DBpedia and Wikidata, of interest in this work. The knowledge hosted by the aforementioned database is customisable according to the domain of application. In this work, the sources integrated in this tool are FrameNet[9] (details on this tool are given in Chapter 5) and Wikidata[5]. Domains are indeed described through the set of basic actions extracted from FrameNet. Each domain element is, furthermore, represented with its characteristics retrieved from Wikidata. Wikidata serves as a human and machine-readable source containing structured data. The Wikidata project has become relevant, to the point that it is being employed as a connecting resource for many different dataset: among others, the Thesauri collected from the Getty Research Institute, such as the Art & Architecture Thesaurus[6] and the Library of Congress[7].

The system studied in this work has a specific application domain, namely the cooking domain. Therefore, all structure-related explanations will be framed in this conceptual area. Details on the structure of the knowledge base, whose peculiarities are employed to search for conflicts, are given in the next section. This pragmatic-related reasoning skill was tested, whose results are reported and discussed. Other important modules of the system, namely speech and intent recognition are tested, as the performances resulting from those modules are important to ensure the proper functioning of the Conflict Search Graph itself.

6.2 Preliminary Performance Analysis

As explained in Chapter 2.2.2, the level at which communicative failures can occur are of four different types: Contact, Perception, Understanding, and Intention. Before analysing how the Common Ground is stored and how inconsistencies are found, it is important to point out what happens at the preceding levels, i.e., speech and intent recognition, where for the first one the acoustic signal is recognised, whereas for the second one the semantic analysis is carried out. In this section, the results of these corresponding processes are show. The data-set used for testing is the SUGAR Corpus, whose structure and content are described in the next section.

[5] https://www.wikidata.org/wiki/Wikidata:Main_Page [last consultation on 19th January 2021]

[6] https://www.getty.edu/research/tools/vocabularies/aat/ [last consultation on 19th January 2021]

[7] https://www.loc.gov/librarians/controlled-vocabularies/ [last consultation on 19th January 2021]

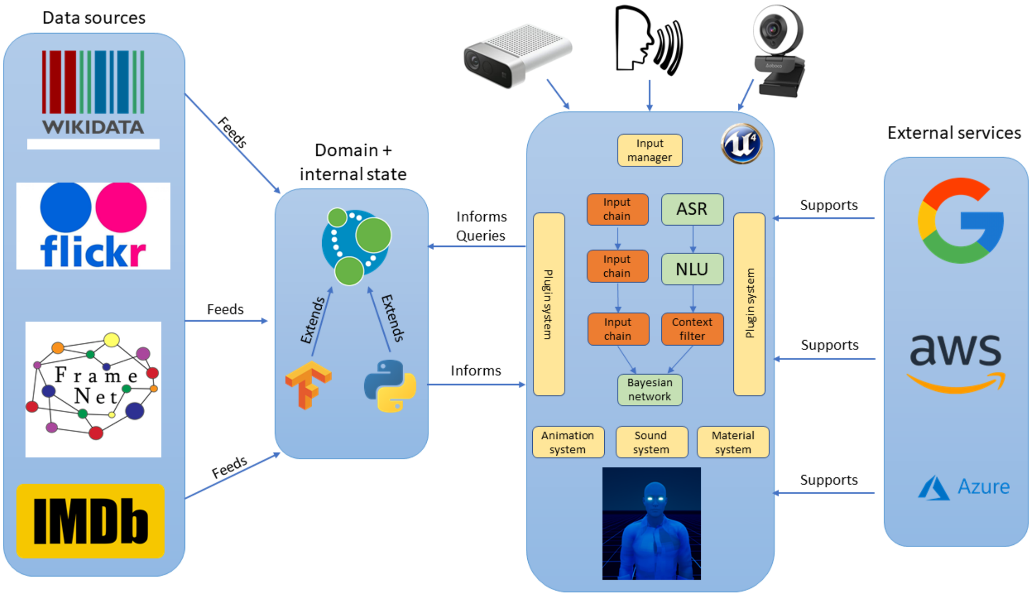


Figure 6.1: The FANTASIA architecture

6.2.1 SUGAR Corpus

The SUGAR corpus[8] was created for EVALITA 2018 [35, 50] and contains 2293 audio files corresponding to Italian cooking actions annotated through predicate-arguments structures [165]. To collect the corpus, a 3D virtual environment was designed. We designed a virtual kitchen in Unreal Engine 4, which could be virtually visited by means of the Oculus Rift[9]. In this kitchen, users could interact with a robot - named Bastian - which received commands to accomplish some recipes, guided via a Wizard-of-Oz. User's orders were triggered by silent cooking videos shown on a TV screen put in the 3D scene, thus ensuring the naturalness of the spoken production. Videos were segmented into elementary portions and sequentially proposed to the speakers who uttered a single sentence after having seen each single frame (Fig. 6.2). The collected corpus thus consists of a set of spoken commands, whose meaning derives from the various combination of actions, items (i.e. ingredients), tools, and different modifiers.

Actions are represented as a finite set of generic predicates accepting an open set of domain-dependent parameters, as follows

$$put(pot, fire)$$

The annotation process resulted in determining the optimal predicate-argument structure corresponding to each command, according to the action templates

[8] Available on GitHub: https://github.com/evalitaunina/SUGAR_Corpus
[9] www.oculus.com

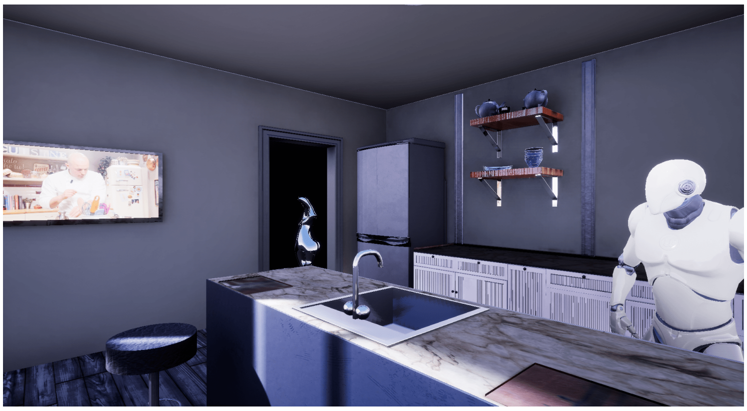


Figure 6.2: 3D Recontruction of Bastian in his Kitchen. On the wall, the television showing frames of video recipes, from which users could extract actions to utter as commands

previously defined through the selected video collection[10] (Table 6.1). In the annotation files, the symbols are univocal: square brackets are used to indicate a list of ingredients, slashes indicate the alternative among possible arguments, asterisks are used when an argument is not explicitly instantiated but recoverable from the context (i.e. previous instantiated arguments, which are not uttered, not even by means of clitics or other pronouns) or from the semantics of the verb (i.e. instrumental verbs). For instance, *fry(flowers)* is represented as *add(flowers, *oil*)* because *oil* is implicitly expressed in the semantics of the verb *to fry* as an instrument to accomplish the action. Among other phenomena, it is worth mentioning the presence of actions paired with templates, even when the syntactic structure needs reconstruction, as in *cover(bowl, wrap)* which is annotated with a more generic template as *put(wrap, bowl)*. In other cases, the uttered action represents the consequence of the action reported in the template, as in *separate(part, flowers)* annotated as *clean(flowers)*, or *stir([yeast, water])* represented with *melt(yeast, water)*.

The arguments order does not reflect the one uttered in the recorded audio files, but the following:

$$action(quantity^{11}, object, complement, modality)$$

The modality arguments are diverse and follow a specific order: *adverb*, *cooking modality*, *temperature* and *time*.

Because of its domain and since users were indeed told they would interact with a virtual agent, this corpus is considered adequate to test our system. Before using this collection as a test set, instead of using the aforementioned predicate-arguments annotation, it has been necessary to re-annotate the data

[10] The videos were selected from the *Giallo Zafferano* website: https://www.giallozafferano.it/

[11] The quantity always precedes the noun it is referred to. Therefore, it can also occur before the complement.

Predicate	**Arguments**
prendere *take*	quantità, [ingredienti]/recipiente *quantity, [ingredients]/container*
aprire *open*	quantità, [ingredienti], recipiente *quantity, [ingredients], container*
mettere *put*	quantità, utensile/[ingredienti], elettrodomestico, modalità *quantity, tool/[ingredients], appliance, modality*
sbucciare *peel*	quantità, [ingredienti], utensile *quantity, [ingredients], tool*
schiacciare *mash*	[ingredienti], utensile *[ingredients], tool*
passare *pass through/strain*	[ingredienti], utensile *[ingredients], tool*
grattare *shred*	[ingredienti], utensile *[ingredients], tool*
girare *turn*	[ingredienti], utensile *[ingredients], tool*
togliere *remove*	utensile/prodotto, elettrodomestico *tool/product, appliance*
aggiungere *add*	quantità, [ingredienti], utensile/recipiente/elettrodomestico/[ingredienti], modalità *quantity, [ingredients], tool/container/appliance/[ingredients], modality*
mescolare *stir*	[ingredienti], utensile, modalità *[ingredients], tool, modality*
impastare *knead*	[ingredienti] *[ingredients]*
separare *split*	parte/[ingredienti], ingrediente/utensile *part/[ingredients], ingredient/tool*
coprire *cover*	recipiente/[ingredienti], strumento *container/[ingredients], instrument*
scoprire *uncover*	recipiente/[ingredienti] *container/[ingredients]*
controllare *check*	temperature, ingrediente *temperature, ingredient*
cuocere *cook*	quantità, [ingredienti], utensile, modalità *quantity, [ingredients], tool, modality*

Table 6.1: Action templates

Frame	Transcription	Frame Elements	Values
Apply_heat	Bastian metti l'impasto in forno	Temperature_setting	-
		Heating_instrument	forno
		Food	impasto
		Container	-
		Duration	-
Cause_to_amalgamate	Sbatti le uova	Parts	-
		Whole	uova
		Means	-
		Place	-
Cause_to_be_included	Metti il fiore di zucca all'intero dell'olio nella padella	Existing_member	olio
		Place	padella
		New_member	fiore di zucca
		Group	-
Cutting	Taglia 15 fiori di zucca	Item	fiore di zucca
		Pieces	-
		Instrument	-
Dunking	Metti i fiori di zucca nella tempura	Theme	fiori di zucca
		Substance	tempura
Grinding	Grattugiare della noce moscata	Instrument	-
		Patient	noce moscata
Placing	Metti l'impasto in frigo per mezz'ora	Theme	impasto
		Area	frigo
		Source	-
		Duration	mezz'ora
Removing	Togli la padella dal fuoco	Source	forno
		Theme	padella
Reshaping	Passa le patate in un passino	Instrument	passino
		Patient	patate
		Result	-
Separating	Togli il pistillo dai fiori di zucca	Whole	fiori di zucca
		Parts	pistillo
		Instrument	-
		Place	-

Table 6.2: SUGAR Annotation Example

using the frame-based one used in the system's knowledge representation. Each command was annotated following the structure of the corresponding frame, among the ones reported in Chapter 5. Examples of frame-based annotations per each intent are given in table 6.2[12].

6.2.2 Automatic Speech Recognition

To test the first step of the system pipeline, the Microsoft Azure Speech Recognition was used. The automatic transcriptions (hypothesis) were then compared to the manual transcriptions (reference) using one of the NIST Scoring Tools, i.e. Sclite (Score Lite) [162]. This was used as a scoring tool not only to get the performance results of the recognition, but also to analytically analyse the possible mistakes, by comparing the distance between reference's words and hypothesis's words. The distance is obtained by finding the cheapest way to transform one string into another. Transformations are of three kinds: i) substitution, which occurs when a word gets replaces; ii) insertion, which oc-

[12] Note that the intents correspond to the Frames considered in Chapter 5 for the recipes annotation

	#Snt	# Wrd	Corr	Sub	Del	Ins	Err	S.Err
Sum	2212	13159	11562	760	837	291	1888	878
Mean	1.0	5.9	5.2	0.3	0.4	0.1	0.9	0.4
S.D.	0.0	3.1	2.7	0.7	1.3	0.4	1.6	0.5
Median	1.0	5.0	5.0	0.0	0.0	0.0	0.0	0.0

Table 6.3: Sclite output showing the number of sentences, number of words, total of correct words, substitutions, deletions, insertions, total of error words, number of sentences with at least one error

curs when a word which was not said is added; iii) deletion, which occurs when a word is left out of the transcription. The numbers of transformations, reported in Table 6.3, were used to compute the Word Error Rate (WER), as illustrated in formula 6.1. The result obtained from this formula is 0.14, which is a good result considering that on average Microsoft API reached 0.18 in the tests reported in [80]. Furthermore, 5.2 words were averagely correctly understood per each sentence, which in turn were made up of 5.9 words on average.

$$\frac{(Substitutions + Deletions + Insertions)}{(Substitutions + Deletions + Correct)} \tag{6.1}$$

However, WER does not account for the reasons why errors may happen and it is not the only important factor for deciding for the accuracy of an ASR. Sclite results helped analysing ASR faults and possible other reasons. In the following sentence, for instance, the recogniser had problems with words segmentation and word recognition.

```
Speaker sentences1837:  1838-raamm27   #utts: 1
id: (1838-raamm27-0000)
File: 1838
Channel: raamm27
Scores: (#C #S #D #I) 6 3 0 0
REF:  quando sono pronti SOLLEVALI E  METTERLI sulla carta
    assorbente
HYP:  quando sono pronti SOLLEVA   LI EMETTE   sulla carta
    assorbente
Eval:                    S         S  S
```

In this example, the substitution of the imperative+clitic form with the infinitive form does not lead to an intent classification error, as the meaning remains the same.

```
Speaker sentences1844:  1845-raamm27   #utts: 1
id: (1845-raamm27-0000)
File: 1845
```

```
Channel: raamm27
Scores: (#C #S #D #I) 7 1 0 0
REF:  e VERSALE in una padella a fuoco basso
HYP:  e VERSARE in una padella a fuoco basso
Eval:   S
```

Similarly, in the following example, a preposition was left out by the recogniser, a mistake which does not affect the intent recognition process. In fact, when testing this utterance in LUIS, the reconised intent is *Apply_heat*. Further details on such tests are given in the next section.

```
Speaker sentences1855:  1856-raamm27   #utts: 1
id: (1856-raamm27-0000)
File: 1856
Channel: raamm27
Scores: (#C #S #D #I) 6 0 1 0
REF:  metti una padella A scaldare sul fuoco
HYP:  metti una padella * scaldare sul fuoco
Eval:                   D
```

Below, as in other cases, an entire sentence extract is left out of the transcription. This happens when the speaker pauses, as the system appears to work as if it was in online mode, where long pauses signal the end of the utterance. The file length does not appear to be relevant to the system. In this case, the recogniser stops the process, as it perceives the input to be over. Specifically, in the example reported here, the silence duration between the first part of the sentence and the rest is of 1.34 seconds. In such situations, *Missing Information CRs* could be possibly used to obtain the missing information and fill the needed slots.

```
Speaker sentences1879:  1880-repam33   #utts: 1
id: (1880-repam33-0000)
File: 1880
Channel: repam33
Scores: (#C #S #D #I) 3 0 13 0
REF:  poi dopo prendi 400 GRAMMI DI BURRO E AGGIUNGILI
HYP:  poi dopo prendi *** ****** ** ***** * **********
Eval:                 D   D      D  D     D D
REF:  AL LATTE CHE HAI VERSATO NELLA PENTOLA
HYP:  ** ***** *** *** ******* ***** *******
Eval: D  D     D   D   D       D     D
```

Finally, in the following example an insertion was found. The adverb *inoltre* was incorrectly segmented and divided in two other existing words. Nevertheless, this mistake should not affect the intent classification, as the affected word is not necessary to the semantics of the intent.

```
Speaker sentences1901:  1902-repam33   #utts: 1
id: (1902-repam33-0000)
File: 1902
Channel: repam33
Scores: (#C #S #D #I) 10 1 0 2
REF:  * nello stesso contenitore aggiungi ** INOLTRE 200 grammi
    di zucchero a velo
HYP:  E nello stesso contenitore aggiungi IN OLTRE   200 grammi
    di zucchero a velo
Eval: I                                   I  S
```

In conclusion, these examples proved that some errors could actually not negatively affect the intent recognition, as the semantic structure can be still correctly retrieved. Furthermore, the application of clarification requests in some contexts can transform recognition mistakes in more natural interactions.

6.2.3 Intent Recognition

Intent Recognition refers to the language understanding task which aims at classifying a user's utterances into one of the predicted categories [164, p. 215]. An *intent* thus corresponds to the user's goal of an utterance in a dialogue session. In this work, the predicted categories correspond to the intents described in LUIS in the form of frames (see also Chapter 5), as reported in the next section, whereas the utterances classified are the actions from the SUGAR Corpus. In Table 6.4, intent per intent F-scores are reported. Two out of eleven intents got no scores because of lack of enough testing data for those classes. The best performing class is *Grinding* with an F-score of 0.97, while the worst performing class is *Placing* with an F-score of 0.54. Nonetheless, many true negatives, such as the ones outputted for the frame *Placing*, were caused by lack of context, as it will be explained in the examples below.

The example 15, labelled as belonging to the class *Apply_heat*, was classified as *Cause_to_be_included* with a confidence of 0.74, as the entity class {Verbs_Cause_to_be_included}, which the verb *metti* belongs to, affected the classification.

(15) *nella prima padella metti a rosolare le patate*

brown the potatoes in the first pan

Frame	**F-score**
Apply_heat	0.85
Cause_to_amalgamate	0.92
Cause_to_be_included	0.74
Cutting	NA
Dunking	0.69
Grinding	0.97
Placing	0.54
Removing	0.60
Reshaping	0.69
Separating	0.67
Storing	NA

Table 6.4: LUIS F-scores intent for intent; NA, which states for *not applicable*, was given to intents with less data

In 16, on the other hand, no intent was assigned to the utterance as no *Ingredient* was mentioned. In this case, the previous context could have helped the classifier. This proves, how semantics alone cannot assure the rightfulness of understanding tasks and how determinant pragmatics is to boost performances.

(16) *accendi il fuoco basso per 10 minuti*

turn on low heat for 10 minutes

In 17, labelled as *Cause_to_amalgamate*, no intent was assigned to the utterance, although with a low confidence (0.45). Here, the verb *battere* was, actually, misused. In the Treccani dictionary, the meanings conveyed by this verb are: i) beat with your hands; ii) beat time; iii) hit in baseball, cricket, tennis, or other ball games; iv) defeat someone; v) insist[13]. On the other hand, the verb is here used with the meaning of its minimal pair, as *sbattere*, that is "to shake, to mix". A lexicon-related clarification request with a disambiguation function could be used in such contexts to solve the problem.

(17) *batti le uova*

beat the eggs

The example 18, was labelled as *Placing.* The predicted label, on the other hand, was *Cause_to_be_included.* These two intents can be confused in some ambiguous contexts. In fact: a) when an *Ingredient* is put in a *Container*, the corresponding intent is *Placing*; b) when an *Ingredient* is put in another *Ingredient* or a set of *Ingredients*, the corresponding intent is *Cause_to_be_included.* Nonetheless, in some utterances, neither the *Container* nor the *Ingredient* are mentioned. To disambiguate between the two intents, the preceding context is

[13] https://www.treccani.it/vocabolario/battere/ [last consultation on 21 January 2021]

important. With the Conflict Search Graph, this is possible, since the sequence of actions is stored. This pragmatic representation is, therefore, essential not only for consistency-seeking purposes, but also to retrieve the aforementioned data implicit in the current state. Furthermore, if the context is still not enough clear, reference-related clarification requests can be adopted.

(18) *metti 30 g di latte intero*

put 30 g of whole milk

In 19, the verb *schiaccia* (En. *press*) misguided the classifier, which assigned the intent *Reshaping* (confidence 0.98) to the utterance labelled as *Dunking*. The verb *schiacciare* belongs instead to the entity class {Verbs_Reshaping}.

(19) *schiaccia nel pangrattato*

press in breadcrumbs

The example 20 was wrongly predicted as *Cause_to_be_included* instead of *Dunking*. The intent template is in fact typical of the predicted intent. Nevertheless, when an *Ingredient* is added to a liquid, the corresponding intent is *Dunking*. The state of the entity can be retrieved from the graph, where the entities are labelled with their state information obtained from Wikidata. This information, representing part of the encyclopedic knowledge, can improve the classification performances.

(20) *mettere il fiore di zucca nella tempura*

put the courgette flower in the tempura

In 21, the utterance was supposed to be classified as *Separating* but no intent was assigned to it, although with a low confidence (0.29). What might have confused the classifier is the denominal verb of removal *sgocciolare* (En. *to drain*). This verb derives from a noun that denotes an object X. In this case, the object *goccia* (En. *drop*) metonynically refers to *water*. The event described by the verb refers to the removal of the object X itself [169]. Other examples of this kind of verbs are *sbucciare* (En. *to peel*), *snocciolare* (En. *to pit*), *spennare* (En. *to pluck*).

(21) *sgocciola le patate*

drain the potatoes

In conclusion, it can be stated that the system performed quite well, as far as both automatic transcription and intent recognition were concerned. Nevertheless, pragmatic approaches comprising the knowledge representation stored in the Conflict Search Graph and the possibility to adopt adequate clarification requests according to the problem, can definitely help the system avoid and/or recover some mistakes. Pragmatics was, therefore, proved to be essential, especially as far as context-related phenomena causing misunderstandings were concerned. For this reason, once speech and intent recognition are processed, the semantic information can be stored in the Conflict Search Graph, whose description and testing phase are the topics of the next section. This graph helps in finding conflicts among the information received in order to motivate and signal them.

6.3 The Conflict Search Graph

The *Conflict Search Graph* is the crucial module of the system, where the knowledge is dynamically stored and checked during the interaction, and where reasoning-like processes occur. The aim of this module is to have a graph where the knowledge domain (i.e., part of the CCG) is stored, and whose conflict search module can be used to signal which input does not respect the rules of the CCG and cannot, therefore, become part of the PCG. In fact, the graph is not just used to represent the domain and its rules: it also supports the automatic process of recognising Common Ground Inconsistencies. Other than detecting unverified preconditions, the graph is used to store the dialogue history so that inconsistencies caused by post-conditions applied by previous actions let the system identify the potential source of the current inconsistency. Pre-conditions of an action describe, in general, the configurations of the CG that are compatible with action instancing. On the other hand, post-conditions are the resulting values assigned to an entity after the action has been processed. When a post-condition resulting from a previous action clashes with a pre-condition of the current action and inconsistency occurs. Whereas the pre-conditions make aware of the possible presence of a conflict, the post-conditions help identify the conflicting action. The check-related process guides the adoption of Clarification Requests (Chapter 2).

The application described in this section, simulates a virtual agent, called Bastian, that accepts commands given in the cooking domain and checks their validity. To build the knowledge base of this application, two main resources were comprised, as previously introduced: Wikidata and FrameNet. From Wikidata, domain elements are retrieved to collect labels and characteristics of the single items involved in the cooking domain. From FrameNet, the set of basic actions involved in the domain is extracted and detailed to support the specific dialogue application. Here, the definition of the domain elements, expressed as SPARQL queries, is presented, together with the frames set and the connecting structure representing the dialogue domain specific for the application. For the cooking domain, represented in the application, specific frame elements were selected, such as semantic roles mainly conveyed by Ingredients, Tools and similar, and connected to Wikidata classes. Besides the data extracted from the aforementioned resources, additional information was added in the graph, namely pre-conditions and post-conditions of specific actions, as it will be illustrated. In this way, whereas from Wikidata not only Italian translation but also item states could be retrieved, from FrameNet action structures are derived. In addition, in the graph, these resources were combined and enriched with pre-conditions rules, as to represent the rule-based structure of the CCG. For example, as a first step, each element labelled as *Ingredient* was defined as an instance of a class descending from the concept *Food* (Q2095) in Wikidata. The set of items representing potential ingredients was obtained using the following SPARQL query

```
SELECT DISTINCT ?item ?itemLabel (group_concat(DISTINCT
?altEN;separator="|") as ?altENs) ?type
{
  {
  ?item wdt:P31 ?class .
  ?class wdt:P279* wd:Q2095 .
  ?item rdfs:label ?itemLabel .

  FILTER(LANG(?itemLabel) = "en")

  OPTIONAL{
    ?item skos:altLabel ?altEN.
    FILTER (lang(?altEN) = "en")
  }

  BIND("instance" AS ?type)
}
UNION
{
  ?item wdt:P279* wd:Q2095 .
  ?item rdfs:label ?itemLabel .

  FILTER(LANG(?itemLabel) = "en")

  OPTIONAL{
    ?item skos:altLabel ?altEN.
```

```
        FILTER (lang(?altEN) = "en")
      }
      BIND("class" AS ?type)
    }
}
GROUP BY ?item ?itemLabel ?altENs ?type
```

Subsequently, the tree-like structure rooted in *Food* was represented in Neo4j and Italian labels were recovered. These steps were performed in separated queries as the number of results was significantly high and timeout errors occurred at the endpoint in this situation. For the representation of other elements of the domain, *Tools* were defined as classes of objects descending from *Kitchen_Utensil* (Q3773693) as in the following SPARQL query

```
    SELECT ?item ?parent ?itLabel ?enLabel
    (group_concat(DISTINCT ?altEN;separator="|") as ?altENs)
    (group_concat(DISTINCT ?altIT;separator="|") as ?altITs) {
  ?item wdt:P279* wd:Q3773693.
  ?item wdt:P279 ?parent.
  ?parent wdt:P279* wd:Q3773693.

  OPTIONAL {
    ?item rdfs:label ?enLabel .
    FILTER(LANG(?enLabel) = "en")
  }

  OPTIONAL {
    ?item rdfs:label ?itLabel .
    FILTER(LANG(?itLabel) = "it")
  }

  FILTER ( bound(?itLabel) || bound(?enLabel) )

  OPTIONAL{
    ?item skos:altLabel ?altEN.
    FILTER (lang(?altEN) = "en")
    }

  OPTIONAL{
    ?item skos:altLabel ?altIT.
    FILTER (lang(?altIT) = "it")
    }
  }
GROUP BY ?item ?parent ?itLabel ?enLabel ?altENs ?altITs
```

Differently from the previous query, instances of classes were not considered as they cover specific objects, like single knives belonging to collections or commercial products. In addition, as the number of results of this query was lower, it was possible to obtain the Italian labels and the tree-like structure in a single query without risking timeout errors. Similarly, *Containers*, were defined as classes descending from the *Tableware* class (Q851782: glasses, plates, etc. . .), *Cooking Instruments* descended from the concept *Cookware_and_Bakeware* (Q154038: cooking pots, casseroles, etc. . .) while *Cooking appliances* descended from the concept *Cooking_Appliance* (Q57583712: stoves, ovens, etc. . .). In Neo4j, the relationships between Wikidata nodes reflect the original ones, as shown in Table 6.5. All imported nodes are provided with the Wikidata ID, the list of English labels, and the list of Italian ones.

Source node	**Relationship**	**Destination Node**
INGREDIENT_INSTANCE	BELONGS_TO	INGREDIENT_CLASS
INGREDIENT_CLASS	SUBCLASS_OF	INGREDIENT_CLASS
TOOL	SUBCLASS_OF	TOOL
CONTAINER	SUBCLASS_OF	CONTAINER
COOKING_APPLIANCE	SUBCLASS_OF	COOKING_APPLIANCE
COOKING_INSTRUMENT	SUBCLASS_OF	COOKING_INSTRUMENT

Table 6.5: Neo4j nodes and relationships

Concerning FrameNet, the entire structure of the resource was modelled in Neo4j following the same labels and relationships available in the original resource. To access the most recent version of FrameNet, online data were collected, rather than using periodic dumps. This was necessary because the dumps offer old versions of FrameNet with no updates. The main Neo4j labels representing the FrameNet structure are FRAME, and FRAME_ELEMENT, which were connected to each other by a BELONGS_TO relationship. For each FRAME and FRAME_ELEMENT, their name was imported, together with frame definitions and related examples.

After organising the base resources in the database, the specific domain structure was established. This served both to connect the original resources and to represent the application-dependent dialogue constraints. First of all, the root of the application-specific domain was represented by a DIALOGUE_DOMAIN node, containing a *name* property to identify the domain. For each of the domain elements recovered from Wikidata, a DOMAIN_ELEMENT node was created, where a *name* property identifies the domain element. In the considered case, DOMAIN_ELEMENT nodes were

Ingredient Tool, *Container*, *Cooking appliance* and *Cooking Instrument*. DOMAIN_ELEMENT nodes were connected to the DIALOGUE_DOMAIN node by BELONGS_TO relationships. DOMAIN_ELEMENT nodes were, then, connected to the Wikidata nodes retrieved using the presented SPARQL queries. As a result, the application-specific domain was connected to Wikidata. During the interaction, users' utterances had to be processed before being checked and stored in the PCG. For this reason, intent recognition was required. The intent recognition system used in Bastian is based on LUIS. An important characteristic of LUIS is that bottom-up information can be provided to support the modelling of relationships between simple entities and more complex constructs built upon sub-entities. Specifically, complex entities can be constructed by declaring simpler entities to be *features* for specific sub-parts of the complex entity. For example, the relationship between a complex entity sub-part representing *Food* and the simple entity *Ingredient* can be made explicit by declaring that *Ingredient* is a feature for *Food*. In this way, upon recognising an *Ingredient* entity, the model can learn faster that the probability of it being assigned to a *Food* sub-entity is higher than other candidates. Also, a sub-list of terms extracted from Wikidata for each of the considered categories was used to initialise the entity recognition module of LUIS, to leverage on the general purpose knowledge the system already has. To keep consistency with the Neo4j representation of the dialogue domain, a LUIS intent was defined for each FRAME_INSTANCE node and a LUIS entity was defined for each DOMAIN_ELEMENT. Then, for each LUIS intent, a complex entity representing the compiled frame was defined. This way, for each example included in the LUIS intents, both the simple entities (e.g., Ingredient, Tool, etc. . .) and complex entities were annotated to let the LUIS's machine learning approach model the relationship between simple entities and complex ones. Furthermore, by using complex entities as *features* for each specific intent, the relationship between complex entities and intents is made explicit to LUIS. The structure of complex entities can be further detailed in LUIS to represent recurring sub-structures that support the recognition of frame elements characterised by specific patterns. For example, in the kitchen domain, specifying a *Tool* size is typical of the *Cutting* frame. Modelling a sub-structure composed by a *Tool* and a *Size* sub-entities lets the LUIS's machine learning approach model the relationship between this specific pattern and the general frame. Figure 6.3 shows an example annotation of the frame *Cutting* matching the structure described by the FRAME_INSTANCE node and the FRAME_ELEMENTs connected to it by a USES relationship. Table 6.8 shows a summary of the annotation levels in LUIS. The same table also shows the features associated to each frame element and, for the case of *Cut-*

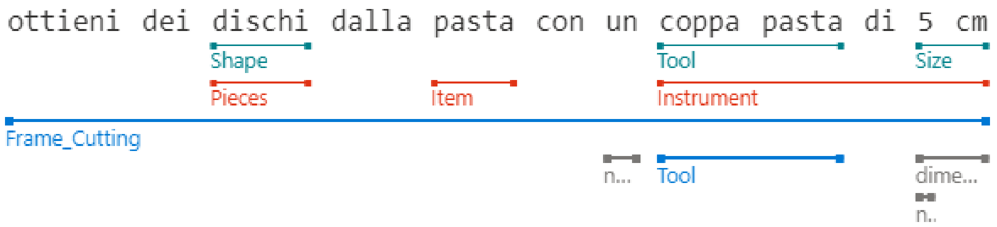

ottieni dei dischi dalla pasta con un coppa pasta di 5 cm

Figure 6.3: An annotated utterance for the frame *Cutting* (En. *Make discs from the dough with a 5cm ring mold*). *Pasta* is a simple entity *Ingredient* and is also an *Item* for the complex entity representing the frame and containing the frame elements as sub-items. In the same way, *coppa pasta di 5 cm* is a *Tool* representing the *Instrument* frame element, of which the specific *size* is specified. *Dischi* occupies the *Pieces* slot although it does not correspond to a simple entity and it specifies a *Shape*.

ting the sub-entities used to provide more details on the sub-structure of the elements associated with a specific frame element.

For each intent, patterns were also defined to better disambiguate between frames, especially when an intent is represented with fewer utterances. For instance, the frames *Separating* and *Removing* can be differentiated, among other things, from each other by the item from which something is moved away. In the former frame, it is an *Ingredient*, whereas, in the latter frame, it is a *Container*, a *Cooking_instrument*, or a *Cooking_appliance*. For this reason, two different patterns where added, as follows[14]

```
{Verbs_Separating} (([il]|[lo]|[la])|([i]|[gli]|[le]))
{Ingredient} (([dal]|[dalle]|[dalla])|[dai]|[dagli]|[dallo]))
{Ingredient}

{Verbs_Removing} (([il]|[lo]|[la])|([i]|[gli]|[le]))
{Ingredient} (([dal]|[dalle]|[dalla])|[dai]|[dagli]|[dallo]))
({Container}|{Cooking_instrument}|{Cooking_appliance})
```

In Table 6.6, the corresponding results containing the specified pattern for the intent *Removing* are shown.

Information coming from Natural Language Understanding and environment perception systems were defined in a specific way to allow standardisation of common ground consistency checks. In the case of deliberation dialogue, a USER node was defined for each human participant. As illustrated in Chapter 3, one peculiarity of this kind of dialogue is that more than two agents can be involved in the exchange; that is also one of the reasons why argumentation-based inference theories cannot be always applied to

[14] Note that articles and complex prepositions followed by an apostrophe in case of elision are missing in the patterns, as LUIS does not allow to concatenate different items' strings when separated by apostrophes

Field	**Value**
Sentence_it	*Togli il composto dal frigo*
Sentence_en	*Remove the mixture from the fridge*
Top intent	Removing
Score	0.994
Entities	composto, frigo
Other entities	{Verbs_Removing: togli}
Top patterns	{Removing}, {Verbs_Removing}

Table 6.6: LUIS detailed results for the utterance *Remove the mixture from the fridge*

Source node	**Relationship**	**Destination Node**
USER	DECLARES	ACTION
ACTION	IS_A	FRAME_INSTANCE
ACTION	REFERS_TO	ENTITY
ENTITY	REFERS_TO	PERCEIVED_ENTITY
ENTITY	ASSIGNED_TO	FRAME_ELEMENT

Table 6.7: Structure of the sub-graph related to ACTIONs

dialogue and, therefore, a dedicated framework is needed. This node thus allows for the representation of each human interlocutor recognised by the systems. ACTION nodes represent declarations from a USER, which is connected to them by DECLARES relationships. Since ACTIONs are always related to FRAME_INSTANCEs, a IS_A relationship was established between ACTIONs and FRAME_INSTANCES they represent. For each recognised ACTION, the linguistic entities recognised in the user utterance were represented by ENTITY nodes coherently with LUIS responses. ACTIONs were linked to ENTITY nodes by REFERS_TO relationships. Moreover, ENTITY nodes were linked to FRAME_ELEMENT nodes, according to the role NLU assigns to the recognised entities, by ASSIGNED_TO relationships. Lastly, objects perceived by the agent in the environment are represented by PERCEIVED_ENTITY nodes, which were linked to DOMAIN_ELEMENT nodes by IS_A relationships. The different types of node separating what is being said from what is perceived are necessary to support *grounding* approaches, where linguistic entities are linked to perceived objects. This also allows to detect inconsistencies between entities present in user utterances and perceived reality. In this case, a simple strategy based on string similarity was used to perform grounding, as the main interest is on conflict detection. The structure of the sub-graph related to ACTIONs is shown in Table 6.7.

Once an ACTION is declared, the related ENTITY nodes are created and linked to the ACTION node by a REFERS_TO relationship. ENTITY nodes are then linked to the PERCEIVED_ENTITY nodes on the basis of the Sorensen-Dice coefficient [152] obtained for every possible pairing between the value property of the ENTITY node and the name property of all the available

PERCEIVED_ENTITY nodes. In this way, plurals, derivative forms, or non-standards forms could be included to be linked to PERCEIVED_ENTITY nodes comprised in the knowledge graph. These are linked to the corresponding PERCEIVED_ENTITY nodes by the relation REFERS_TO. Nodes and relationships were generated using the following Cypher query:

```
MATCH (a:ACTION) WHERE NOT (a)-[:IS_FOLLOWED_BY]->()
WITH a
MATCH (pe1:PERCEIVED_ENTITY), (e:ENTITY)<-[:REFERS_TO]-(a)
OPTIONAL MATCH (pe1)<-[:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)
WITH pe1, a, e, pe2, COLLECT(pe2)[0] AS successor
WHERE successor IS NULL OR NOT successor.name = pe1.name
UNWIND split(apoc.text.replace(e.value, "\[[\.\d]+\]", ""), ",") AS names
WITH pe1.name AS name, COLLECT(names) AS names,  apoc.text.sorensenDiceSimilarity(names, pe1.
    name) AS score, a
WITH MAX(score) as maxValue, a
MATCH (pe1:PERCEIVED_ENTITY), (e:ENTITY)<-[:REFERS_TO]-(a)
OPTIONAL MATCH (pe1)<-[:CREATEFROM]-(pe2:PERCEIVED_ENTITY)
WITH maxValue, pe1, a, e, pe2, COLLECT(pe2)[0] AS successor WHERE successor IS NULL OR NOT
    successor.name = pe1.name UNWIND split(apoc.text.replace(e.value, "\[[\.\d]+\]", ""), ",")
     AS names
WITH pe1.name AS bestMatch, COLLECT(names) AS names, COLLECT(apoc.text.sorensenDiceSimilarity(
    names, pe1.name)) AS score, maxValue, a
WITH bestMatch, apoc.coll.zip(names, score) AS pairs, maxValue, a
WITH bestMatch, MAX([pair IN pairs WHERE pair[1] = maxValue])[0][0] AS entityName, a
WHERE entityName IS NOT NULL
WITH entityName, bestMatch, a
MATCH (pe1:PERCEIVED_ENTITY), (e:ENTITY)<-[:REFERS_TO]-(a) WHERE pe1.name = bestMatch AND e.
    value CONTAINS(entityName)
OPTIONAL MATCH (pe1)<-[:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)
WITH entityName, a, pe1, e, pe2, COLLECT(pe2)[0] AS successor WHERE successor IS NULL OR NOT
    successor.name = pe1.name
CREATE (pe1)<-[:REFERS_TO {label: entityName}]-(e)
```

Listing 6.1: Query relating an ENTITY to a PERCEIVED_ENTITY

To connect the dialogue domain to FrameNet, a similar strategy was adopted. In total, 10 frames were used in the presented application: for each of these frames, a FRAME_INSTANCE node was created and connected to the original FRAME by an INSTANCE_OF relationship. Also, for each frame, a subset of FRAME_ELEMENT nodes was considered for the application domain. To represent this, a USES relationship was established between the FRAME_INSTANCE node and the FRAME_ELEMENT node of interest. To indicate which domain elements can be associated with a FRAME_ELEMENT in the application domain, CONSTRAINT nodes were established. First of all, FRAME_INSTANCE nodes were connected to CONSTRAINT nodes by a HAS_CONSTRAINT relationship. Then, the CONSTRAINT node was connected to the FRAME_ELEMENT node it was applied to by a REFERS_TO relationship and to a DOMAIN_ELEMENT node that can be associated to the FRAME_ELEMENT by another REFERS_TO relationship. CONSTRAINT nodes can, therefore, be used to describe which DOMAIN_ELEMENTS can be associated to fill a slot based on a FRAME_ELEMENT in a dialogue management system. Since Framenet does not provide pre-conditions and post-conditions for the application of the related actions, these must be defined at application level: in this case, pre- and post-conditions are represented as properties of the FRAME_INSTANCE nodes and contain Cypher queries designed to verify, given the way the specific application manages common ground updates, that the necessary checks are performed before accepting a user-declared action. To be interpreted by a single function, in the application logic, the results format is constrained to a table containing a row for each pre-condition to be tested. Each row consists of the following columns:

- Eval: the truth value of the pre-condition;
- ConflictingAction: the ID of the ACTION node causing a pre-condition to be violated, if present
- NLExplanation: a fragment of text providing an explanation, in natural language, of the violated pre-condition;
- ConflictingFrame: the name property of the FRAME instanced by the FRAME_INSTANCE causing the conflict;
- OriginalEntity: the name property of the PERCEIVED_ENTITY involved in the ACTION causing the violation.

As a pre-condition example, consider the *Grinding* frame. As showed in Listing 5 in Appendix B, the FRAME_ELEMENT *Patient* is checked with the UNION of three separated sub-queries, each considering a different pre-condition, to verify that it is not populated with an entity, whose quantity is no longer available, or with an entity which is is neither liquid nor already in a powder form.

Running this query on a graph representing the common ground configuration is, thus, important to check whether the last ACTION can be accepted or not, in that it is verified that the updated graph does not violate the pre-conditions set by the activated FRAME_INSTANCE. Figure 6.4 shows the application level dialogue domain as an intermediate graph structure connecting the knowledge provided by Wikidata and FrameNet.

If all pre-conditions are verified, the declared ACTION can be accepted and post-conditions can be applied. For the case of the FRAME_INSTANCE related to the FRAME *Grinding*, the PERCEIVED_ENTITY related to the ENTITY assigned to the *Patient* FRAME_ELEMENT becomes a new version of itself, which acquires the POWDER label. The *Grinding* post-conditions

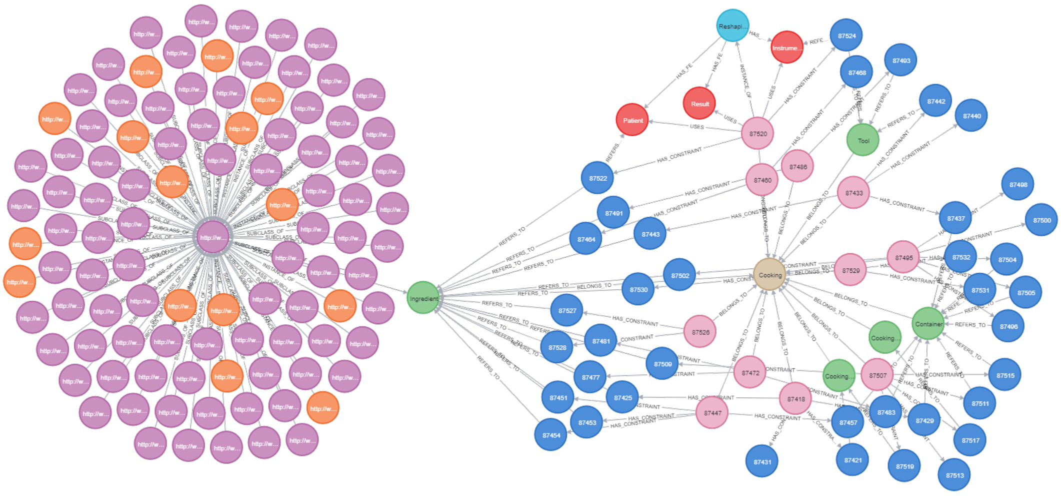


Figure 6.4: The application level dialogue domain connecting Wikidata and FrameNet. The structure of the original resources is preserved in this schema while the dialogue domain sructure and constraint inform the served application. Purple and orange nodes represent Wikidata instances and classes, green nodes represent DOMAIN_ELEMENTs, blue nodes represent CONTRAINTs, red nodes represent FRAME_ELEMENTs, pink nodes represent FRAME_INSTANCEs. For illustration purposes, only one FRAME node (in cyan) is reported. The brown node represent the DIALOGUE_DOMAIN node.

are declared as in Listing 9 in Appendix B. The pre-conditions defined before would not be verified now, for the most recent version of the involved PERCEIVED_ENTITY. This is because it cannot be assigned to the *Patient* FRAME_ELEMENT for an ACTION related to the FRAME_INSTANCE referring to the FRAME *Grinding*. The Neo4j graph representing a user utterance and its role in the common ground is shown in Figure 6.5.

To connect the internal knowledge representation hosted in Neo4j with the probabilistic framework provided by LUIS, the FANTASIA framework is used. The Neo4j module provides access to the graph-based representation of the CG and to the dialogue history. The Azure module provides access to the remote services to perform ASR, intent/entity recognition and TTS. UE4 manages the interaction using the 3D interface and the information provided by the other modules. The system architecture obtained by deploying FANTASIA in the cooking domain is shown in Figure 6.6. UE4 also hosts the application logic, generating the virtual agent's behaviour using an underlying model based on the results presented before. To allow updates to the domain representation to be reflected in UE4, the system first queries the graph database to obtain the list of FRAME_INSTANCEs and their CONSTRAINTs, dynamically initialising internal data structures to match the ones obtained from Neo4j. These are used in UE4 to support the creation of appropriate queries once user utterances are analysed. After obtaining a structured representation of the user's utterance from the LUIS backend, the CG manager matches the intents and en-

	Frame elements	**Sub-Entities**	**Features**			**Frame elements**	**Sub-Entities**	**Features**
Apply heat	Temperature setting	-	Temperature		Cause to amalgamate	Parts	-	Ingredient
	Heating instrument	-	Cooking appliance			Whole	-	Ingredient
	Food	-	Container			Means	-	Tool
			Ingredient			Place	-	Container
	Container	-	Cooking instrument				-	Cooking instrument
	Duration	-	Duration					
Cause to be included	Existing member	-	Ingredient		Cutting	Item	-	Ingredient
	Place	-	Container			Pieces	Quantity	Number
	New member	-	Ingredient				Size	Dimension
							Shape	-
	Group	-	Ingredient			Instrument	Size	Dimension
							Tool	Tool
Dunking	Substance	-	Container		Grinding	Instrument	-	Tool
		-	Ingredient			Patient	-	Ingredient
	Theme	-	Ingredient					
Placing	Theme	-	Container		Removing	Source	-	Container
		-	Cooking instrument				-	Cooking appliance
		-	Ingredient				-	Cooking instrument
	Area	-	Container					
	Source	-	Container			Theme	-	Container
		-	Cooking appliance				-	Cooking instrument
		-	Cooking instrument					
	Means	-	Tool				-	Ingredient
	Duration	-	Duration					
Reshaping	Instrument	-	Tool		Separating	Whole	-	Ingredient
	Patient	-	Ingredient			Parts	-	Ingredient
	Result	-	-			Instrument	-	Tool
						Place	-	Container

Table 6.8: Frame structures with features and sub-entities in LUIS.

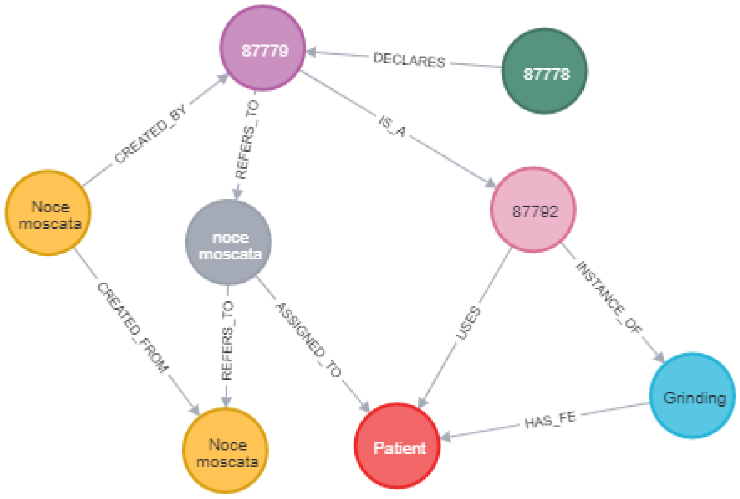


Figure 6.5: The graph representing the relationship between data coming from an NLU system in the common ground given the user utterance *Trita la noce moscata* (Grind the nutmeg). A USER (green) DECLARES an ACTION (purple), which IS_A FRAME_INSTANCE (pink) of the FRAME (cyan) *Grinding*. The ACTION REFERS_TO an ENTITY (grey), that is assigned to the FRAME_ELEMENT (red) *Patient* of *Grinding* and REFERS_TO a PERCEIVED_ENTITY (yellow). According to the *Grinding* post-conditions, a second PERCEIVED_ENTITY is CREATED_FROM the original one representing the *noce moscata*. The new PERCEIVED_ENTITY is also CREATED_BY the ACTION and it has the POWDER label.

tities detected by LUIS with, respectively, frames and FRAME_ELEMENTs, as described in the previous subsection. To simulate the process of *hypothesising* the situation after accepting the ACTION resulting from the analysis of the user utterance, the CG manager opens a transaction in Neo4j, adding the ACTION and its related structure without committing changes. This way, it is possible to work with a volatile version of the updated database that can be easily rolled back, should the ACTION be rejected. In this way, a *hypothetical common ground* is created to check for consistency based on the rules defined in the graph. Pre-conditions are, therefore, checked inside the open transaction and the graph database compiles a report following the structure previously described. The CG manager, using this information, commits the changes together with post-conditions if all pre-conditions are verified and generates an acknowledgement utterance to be synthesised by the TTS system. If a pre-condition is not verified inside the transaction, the changes are rolled back and the data included in the Neo4j report are used to generate an appropriate feedback message: in this case, a negative polar question. In other words, given the sequence of frames activated by user utterances $F = \{f_1, ..., f_k\}$, for each argument of the current predicate evoking a specific instantiated semantic frame f_k, and given the preconditions $s-pre_k = \{p_1, ..., p_n\}$ of the $k-th$ frame, when p_i of the semantic role of that argument is verified for $1 <= i <= n$, no conflicts arise.

If a conflict occurs, it must be signalled in order to enable subsequent repair. The fact that pre- and post-conditions are explicitly reported in the graph is not only useful to find the conflict, but also to explain why an action cannot be accepted, possibly indicating the source of the error. Before highlighting the conflicting action with a polar question, the system explains why the action cannot be performed. For instance, if the user asks the system to grind an ingredient which was already ground in a previous action, Bastian will reply with *I can't. X is ground* followed by the question *Didn't I have to grind X?* The data building the explanation are retrieved from a Cypher query and specifically from the aforementioned NLExplanation column. The explanation given here is of the type *why*-explanation, which is used to convey the underlying, hidden reasons for an action or event [157]. While explanations are a possible other signal of artificial A-Consciousness (Chapter 3) and are found to increase the understandability and desirability of agents' behaviours [157], they can be cause of failures in case of inconsistencies. Although explanations are useful in the interaction, as they undo the devastating consequences of logical inconsistencies, they are not sufficient to detect the conflict [81]. As demonstrated in the previous chapter (Chapter 5), the capability to identify the source of the error is increased when negative polar questions are adopted. The combination of both explanations and clarification requests can, therefore, consistently improve the interaction. If, on the other hand, the ACTION can be accepted, the NL feedback generated is a simple feedback with an *Acknowledgement* pragmatic function [136]. The system logic flow is summarised in Figure 6.7.

6.3.1 Conflict Search Graph Experiment

The configuration of the graph, at this point, represents the Common Ground and contains all information concerning dialogue history, shared knowledge, consistency checks, and action consequences. It is, therefore, possible to use the Cypher query language to describe the pre-conditions under which an action can be performed and, similarly, the graph updates following the acceptance of the action itself. The standardised format of the pre-conditions query output and the representation of the queries themselves as FRAME_INSTANCE properties support, from the application logic point of view, a universal way to manage the process of retrieving pre-conditions queries, running them to obtain consistency checks and applying post-conditions.

In Appendix B, queries checking pre-conditions and describing post-conditions for different frames are displayed. Here, a deeper presentation is presented for a few selected frames for illustration purposes. Listing 2 represents the pre-conditions of the frame *Cause_to_amalgamate*: the query checks only

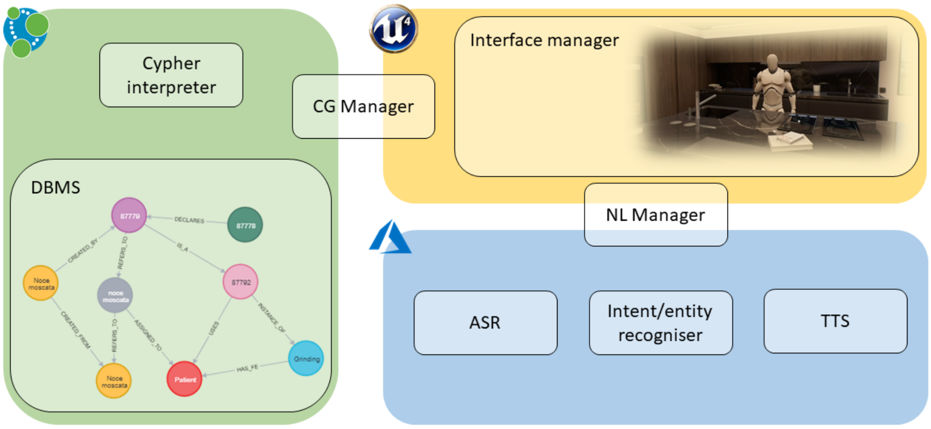


Figure 6.6: The Bastian architecture obtained by deploying FANTASIA in the cooking domain. The CG Manager connects UE4 to the Neo4j module while the NL Manager connects UE4 to the NLU and TTS services provided by Azure. UE4 hosts the application logic and manages the interaction through the interface.

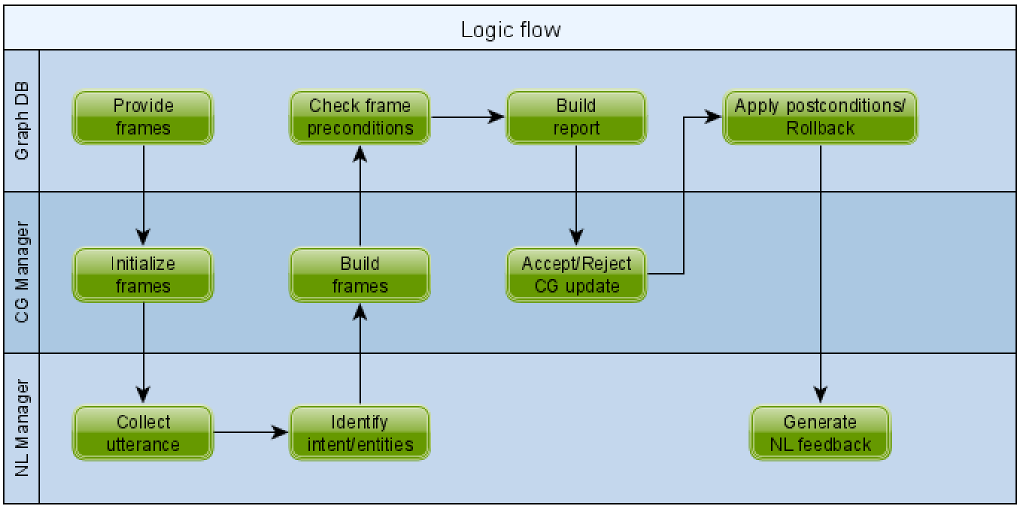


Figure 6.7: The logic flow in the cooking domain

the condition that a sufficient amount of the target PERCEIVED_ENTITY nodes, to which ENTITY nodes REFER_TO, is available. The checks are performed in the *hypothetical common ground*, as mentioned before. First of all, the last ACTION in the sequence, defined as the only ACTION that is not FOLLOWED_BY another ACTION, is retrieved together with the ENTITY nodes it REFERS_TO. From these ENTITY nodes, the corresponding PERCEIVED_ENTITY nodes they REFER_TO are retrieved. To compute the available quantity of the ingredient represented by each PERCEIVED_ENTITY, the PERCEIVED_ENTITY nodes that have been CREATED_FROM the currently targeted PERCEIVED_ENTITY nodes are also retrieved, if present. The available quantity is, therefore, computed by subtracting from the initially available quantity, recorded in the target PERCEIVED_ENTITY node, the quantity used by the PERCEIVED_ENTITY nodes CREATED_FROM the target one. Next, the quantity used by the currently declared ACTION must be considered. If the quantity was declared by the user, it is directly considered in the rest of the evaluation. If a quantity was not specified, the quantity to be used corresponds to the remaining quantity, as computed before. The pre-condition is verified if, by subtracting the already used quantities and the quantity involved in the current ACTION, the result is at least 0. A special case is considered when a PERCEIVED_ENTITY is considered to be available in $Infinite$ quantity, as in the case of spices. To avoid an Infinity - Infinity operation, which would return NaN, an undeclared quantity defaults to 1 so that 1 - Infinity is still greater than 0. If the pre-condition is not verified, the information found in the other fields of the response contains the data to build the error message, consistently with the $B(p)\_E(\neg p)$ condition. Specifically, the NLExplanation contains the string *Non ho abbastanza X* (En. *I don't have enough X*) where *X* is the label of the PERCEIVED_ENTITY violating the pre-condition. To generate the HNPQ, the data of an ACTION, which previously used the same PERCEIVED_ENTITY, are extracted, such as the ID, the name of the frame, and the list of the ingredients involved. In UE4, then, these data are concatenated with the preceding string *Non dovevo* (*Didn't I have to*) to build the request. In Figure 6.8 the quantity-related conflict for the frame *Cause_to_amalgamate* is shown. Here, the ACTION *mescola il burro con il latte* (En. *mix butter with milk*) uses all the available PERCEIVED_ENTITY as no quantity was specified. Consequently, the following ACTION *mescola la farina con il latte* (En. *mix flour with milk*) does not verify the pre-condition, because there is no more milk to be used. In addition, the responsible ACTION is identified with the preceding one, so that the related data can be used to generate the error message containing both the general explanation and the HNPQ.

In Listing 5 in Appendix B, the pre-conditions for *Grinding* are checked. Other than checking that there is a sufficient quantity of the target PERCEIVED_ENTITY available (Condition 1), the query also checks that the target PERCEIVED_ENTITY does not have the POWDER (Condition 2) and the LIQUID (Condition 3) labels, as it is not possible to grind powder and liquids. The query is composed by the UNION of three different queries each checking a different condition, among the considered ones. The assembled output results in a table having a row for each pre-condition being checked, so that it is possible to reconstruct exactly which conditions were verified and which ones were not, during the generation of the feedback. To detect the possible cause of the inconsistency, the query backtracks the chain of CREATED_FROM relationships starting from the PERCEIVED_ENTITY violating the pre-conditions and checks for the presence of a previous version of the target PERCEIVED_ENTITY that did not have the POWDER/LIQUID label. If this pattern can be matched, the ACTION that REFERS_TO an ENTITY which, in turn, REFERS_TO the PERCEIVED_ENTITY without the POWDER/LIQUID label is responsible for introducing the label and is reported by the query as the possible cause of error. In Figure 6.9, the PERCEIVED_ENTITY *noce moscata* (En. *nutmeg*, acquires the label POWDER as an effect of the post-condition of the first *Grinding* ACTION. When the USER DECLARES a new ACTION that, again, attempts to grind the nutmeg, the presence of the POWDER label makes Condition 2 to be unverified. As a consequence, the error message describes the problem and indicates the preceding ACTION as responsible for the inconsistency by means of an HNPQ.

To verify that the Conflict Search Graph structure could actually assure the detection of inconsistencies to be, consequently, properly signalised, dedicated tests were carried out. For the test, 20 recipes were used, 10 of which corresponded to the ones used for the experiment described in Chapter 5. In addition to them, 10 more recipes were used, which were excluded in the preceding dialogue modelling phase. The corresponding ACTIONs of each recipe are reported in Appendix A, where the conflicting inputs are signalised in bold. In table 6.9, the test results are displayed. For three recipes, namely *Pancakes*, *Piadina Romagnola*, and *Polpettine*, the expected conflict action did not correspond to the one selected by the system. Nevertheless, the system outcomes cannot be considered as proper mistakes, as the system choices have a reasonable explanation. For instance, for the Pancakes recipe, the manually inserted conflict corresponded to *melt butter in a pan*, where no quantity was specified. The conflict is triggered when the last action *put the butter in the pan* is received in input, as the butter is no longer available. Nonetheless, the conflict was found at a_5, *add milk and butter to the yolks*. In a sense, whereas

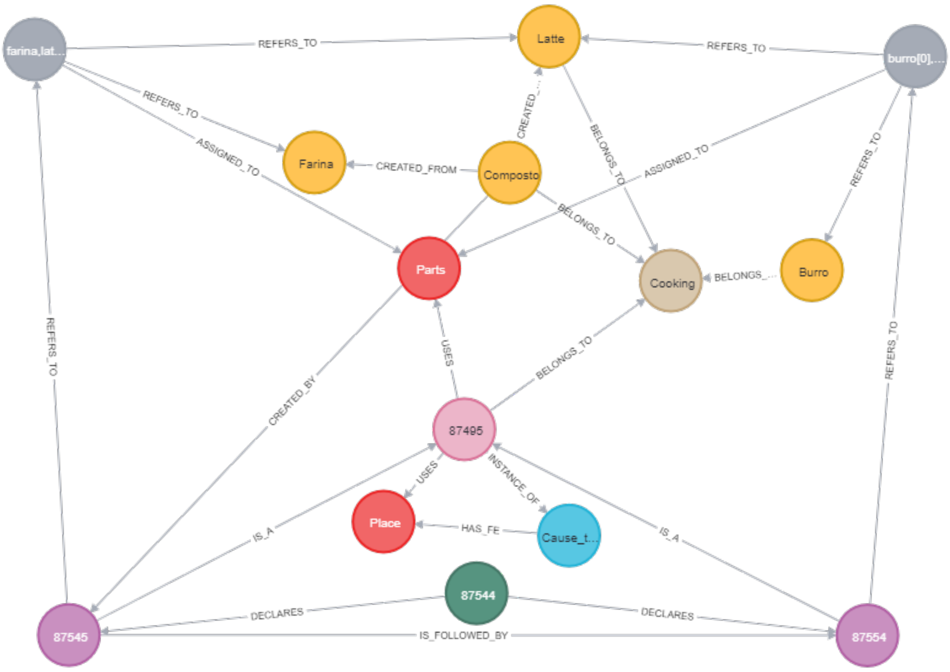


Figure 6.8: Quantity-related conflict

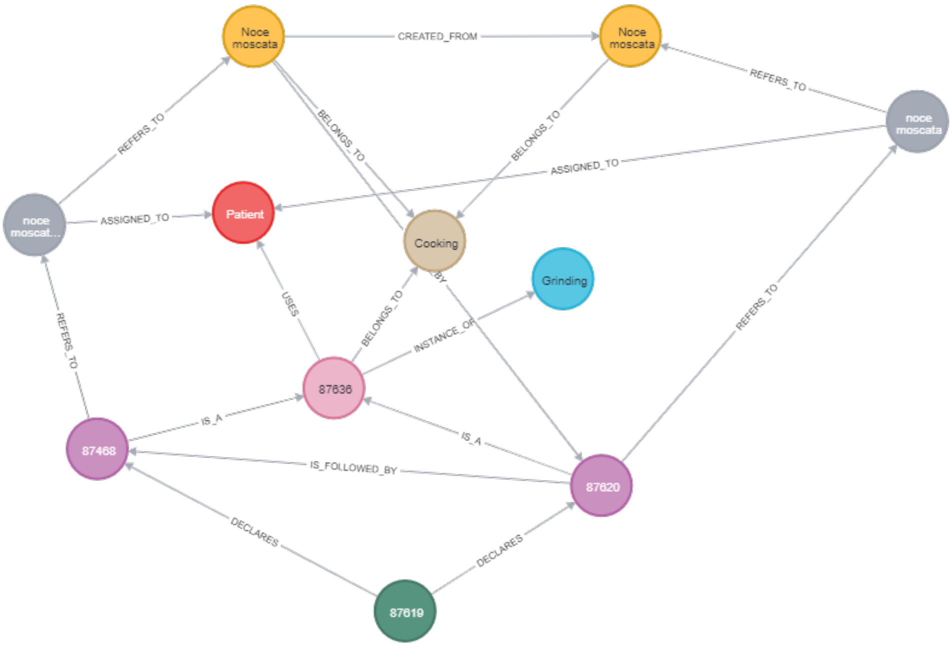


Figure 6.9: Grinding conflict

the whole butter quantity was used in the action of melting it, it is only at a_5 that the butter was actually used in combination with other ingredients, an action which makes it no longer available to be put in a pan. In the subsequent operation 6, the butter is then mixed with the other ingredients, and it is here that the reference to butter is completely lost. The fact that only one action was expected as a result is given by the fact that, in the experiment explained in Chapter 5, participants had to identify the conflict caused by the first use of the item causing the problem. Similarly, in the Piadina recipe, the conflict was inserted at a_1 by replacing *put part of the flour in the bowl* with *put the flour in the bowl*. The conflict is triggered when the operation "dust the work surface with flour" is received in input, as the flour is no longer available. The system found the conflict at a_2 *add lard, salt, baking soda and little water to the flour*. As before, the only constraint required is the usability of flour, which at a_2 was still available and stopped being usable after being mixed with other ingredients. Furthermore, it had not yet undergone any change of status. The next action, corresponding to *Cause_to_amalgamate*, is where everything is mixed and where the reference to the flour is lost. Finally, for the Polpettine di tonno recipe, the ingredient *ricotta* (*add parmesan, tuna, eggs, and anchovies to the ricotta*) was replaced by *breadcrumbs* (*add parmesan, tuna, eggs and anchovies to breadcrumbs* in a_4. The conflict was found by the system in a_5 where other ingredients were added to the *breadcrumbs*, making the breadcrumbs no longer available. This ingredient was, in fact, needed in a subsequent action, where meatballs had to be dunked in it. These results proved that the system could, therefore, analyse pre-conditions rules correctly in a real context of use, in a way that was even more precise than expected by designers.

Recipe	**Result**	**Expected Result**	**Outcome**
Besciamella	3	3	OK
Carbonara	10	10	OK
Cestini ripieni	7	7	OK
Crocchette	5	5	OK
Pancakes	5	1	KO
Patate al forno	4	4	OK
Piadina romagnola	2	1	KO
Pizzette rosse	3	3	OK
Polpettine di tonno	5	4	KO
Tiramisù	6	6	OK
Gnocchi	6	6	OK
Guacamole	6	6	OK
Hamburger di ceci	5	5	OK
Mousse al cioccolato	9	9	OK
Plumcake	1	1	OK
Polpette di zucchine	5	5	OK
Sformato di verdure	7	7	OK
Torta Tenerina	6	6	OK
Zucchine alla scapece	4	4	OK
Zuppa	7	7	OK

Table 6.9: Conflict Search Graph Results and Outcomes

Chapter 7

Conclusions

In Human-Machine interaction, the study and application of pragmatic aspects has interested few phenomena, although their importance was recognised in different studies. Error handling and requests for clarification have always had a central role, since the correct understanding and the consequent task completion of the system are the desired goals. If commercial systems try to identify possible mistakes which can be caused by users or by technology limits, their ability of understanding the real cause of problems to adequately signal them and let the human user correct them is still a frontier to be explored. Among the pragmatic tools considered to handle errors, Clarification Requests are the most frequently used. Such communicative tools, when adopted by automatic dialogue systems, generally deliver confirmation and check of the correctness or completeness of slot-filling processes. Moreover, Clarification Requests tend to refer to the utterance that precedes them. Interestingly, even when the current utterance can be perfectly correct when received as an input, in a subsequent moment, the same utterance can be successively re-evaluated as an error, as the grounded information can clash with a newly introduced contextual evidence. In this work, the use of Clarification Requests handling Common Ground inconsistencies was investigated. In this perspective, on the one hand, the study of Clarification Request forms was needed, and, on the other hand, a system capable of managing Common Ground and, therefore, dialogue history, had to be designed.

The complexity of possible misunderstanding and conflicting situations makes it necessary to study the communicative strategies used to efficiently handle the related interaction problems. Since different error-related pragmatic needs could be expressed by diverse syntactic forms, an inquire about the syntax-pragmatics interface was needed. In Chapter 3, the analysis of the SaGA Corpus resulted in the presence of frequent Information Processing problems, among which Common Ground Triggers were the most numerous. Furthermore, positive and negative polar questions resulted to be more frequently used in Common Ground conflicting scenarios, compared to Missing Information problems. To better understand what kind of form was the most appropriate according to the pragmatic need, the experiment described in Chapter 4 was carried out. The aim of the experiment was to compare the results obtained from the corpus analysis of semi-spontaneous dialogues with those provided by a pre-constructed situation as that presented in [56]. The results generally confirmed the tendencies that the annotation anticipated. In particular, as far as Italian was concerned, when the original bias clashes with a contextual evidence, that is when a Common Ground Inconsistency occurs, the high negation polar question in the past tense was the most frequently selected form. The role of different forms of polar questions was investigated in the experiments described in Chapter 5, which mainly considered the use of both positive and negative polar questions with respect to general error messages in human-machine interaction to prove that the use of a particular form could improve robustness. The use of high negation polar questions in the past tense had a consistent impact on conflict understanding and resolution, confirming their importance. In fact, the experiments demonstrated that these questions were more efficient in helping the subjects understand the errors and solve them faster. Such questions were, therefore, used as grounding feedback when inconsistencies of this kind occurred in human-machine interaction. In Chapter 6, an argumentation-based dialogue system architecture was presented. Such a system used a graph-based representation of the Common Ground, to highlight possible conflicts, in order to signal them in an efficient way, such as with a Clarification Request in the form of a high negation polar question in the past tense. The system's dialogue manager was designed to react to problems which are not immediately evident, but which could depend on the history and state of the dialogue. An erroneous action caused either by the system or the user, when consistent with the Common Ground, can still be accepted. It is only at a later stage of the dialogue that the error can become clear. In such situations, a similar system should be designed to trace back the actions, find the inconsistency, and report it in an appropriate way.

To sum up, the four research questions presented in Chapter 1 were answered as follows

RQ1 *Which forms of Clarification Requests are frequently adopted by speakers when Common Ground Inconsistencies occur?*

Polar questions resulted to be the most frequent form of Clarification Requests adopted by speakers when Common Ground Inconsistencies occur. Among them, high negation polar questions in the past tense are the most frequent ones. These, in fact, are considered as the most appropriate forms to function as *epistemic vigilance* tools. Specifically, they express an epistemic bias towards a grounded presupposition which finds a contradiction in the contextual evidence. This was proved both through a corpus-based analysis and a linguistic-based experiment.

RQ2 *Do Common Ground Inconsistencies require specific superficial polar question forms in Italian as well as in English and German?*
Beside the conflicts between a positive bias and negative contextual evidence, other types of conflict can cause the adoption of different other forms of polar questions. As in [56], this was also demonstrated for Italian.

RQ3 *Does using a specific polar question form result in an improved communication efficiency, or is it just a matter of naturalness?*
The adoption of high negation polar questions in the past tense was proved not to be just a matter of perceived appropriateness and consequent naturalness, but it also resulted in an improved communication efficiency. The adoption of such question forms rather than positive polar questions or general error messages helped the user find and solve conflicts more frequently.

RQ4 *How can Common Ground Inconsistencies be detected and signalised in computational architectures for dialogue management systems?*
Common Ground Inconsistencies can be detected and solved in computational architectures for dialogue management systems by adopting Common Ground graph representations based on semantic resources, such as Wikidata, for lexical and conceptual information, and FrameNet for action structures. Furthermore, the addition of domain-dependent CCG rules, in the form of Cypher queries, are applicable as dialogue manager support to find possible conflicts. When conflicts are found, explanations are synthesised to make the user aware of the presence of a conflict. In addition, when conflicts depend on an action previously given in input by the user, Clarification Requests are adopted to signal the cause of the conflict itself. The proposed organisation of the data in the graph allows an external application, implementing human-machine interaction logic, to handle the action interpretation and validation cycle in a general way.

As mentioned at the beginning of this work, the aim pursued here was to increase the number of investigations and applications of pragmatics in conversational agents. In fact, in the last ten years, semantics has been a more investigated topic within the dialogue systems field with respect to pragmatics. Moreover, despite the fact that Clarification Requests are one of the grounding tools used in conversation, their study and application in dialogue systems have not yet seen a boost. An in-depth analysis of pragmatic phenomena related to Common Ground construction and consistency checks in human-machine interaction, with the use of Clarification Requests, was therefore missing. Concerning the form of Clarification Requests, although in this thesis the role of high negation polar questions in signalling conflicts of the $p \neg p$ type has been investigated, the studies conducted in [56] and extended in this work show that there are patterns that link pragmatic situations, described in terms of contrast between bias and contextual evidence, to syntactic forms. The impact of all those forms can be studied in the corresponding cases for future development.

In this study, Clarification Requests are investigated and adopted on the basis of context analysis, determinant in establishing when a command is incoherent, thus in a $B(p)_E(\neg p)$ scenario. For context analysis here is intended the storing of consecutive commands and the possibility to conceptually relate them. This information is represented in the form of a graph. The use of graph databases as substitutes for classical reasoning engines opens up important developments in both the representation of information and the use of hybrid systems, since support is provided for both machine learning algorithms and rule-based systems which search for patterns in the graph. System performance can further benefit from methods such as query parameterisation, caching, and index building, which are not available in reasoning engines. According to what Prakken recently wrote [120], there is still no theoretical framework comparable to that existing for argumentation-based inference for the case of argumentation-based dialogue. The work presented here constitutes a first exploration of a Common Ground representation methodology for the detection of conflicts, a process which is fundamental in Argumentation-based dialogue and which opens the possibility of providing a formalisation of the problem based on graph configurations.

Appendices

Appendix A

Recipes

1) Besciamella

Apply_heat Food:burro;Container:pentola;
Grinding Patient:noce moscata;
Cause_to_be_included New_member:noce moscata;Existing_member:burro;
Cause_to_be_included New_member:part#latte;Existing_member:burro;
Cause_to_amalgamate Parts:burro;
Cause_to_be_included New_member:farina;Existing_member:composto;
Cause_to_amalgamate Parts:composto;
Apply_heat Food:latte;Container:pentolino;
Cause_to_be_included New_member:noce moscata,sale;Existing_member:latte;

2) Carbonara

Apply_heat Food:acqua,sale;Container:pentola;
Removing Source:guanciale;
Cutting Item:guanciale;
Placing Theme:guanciale;Goal:padella;
Apply_heat Food:guanciale;Container:padella;
Storing Theme:guanciale;Location:da parte;
Cause_to_be_included New_member:spaghetti;Existing_member:sale,acqua;
Apply_heat Food:spaghetti;
Placing Theme:tuorli;Goal:ciotola;
Cause_to_be_included New_member:guanciale;Existing_member:tuorli;
Cause_to_be_included New_member:pepe;Existing_member:tuorli;
Cause_to_amalgamate Parts:tuorli;
Cause_to_be_included New_member:acqua di cottura;Existing_member:composto;
Cause_to_amalgamate Parts:composto;
Removing Source:spaghetti;
Cause_to_be_included New_member:spaghetti;Existing_member:guanciale;

3) Cestini ripieni

Removing Source:datteri;
Placing Theme:datteri;Goal:mixer;
Cause_to_amalgamate Parts:datteri;Whole:crema;
Placing Theme:crema;Goal:ciotola;
Cause_to_be_included New_member:cannella;Existing_member:crema;
Cause_to_be_included New_member:sale;Existing_member:crema;
Cause_to_be_included New_member:yogurt greco;Existing_member:crema;

Cause_to_amalgamate Parts:composto;
Cause_to_be_included New_member:fiocchi d'avena;Existing_member:composto;
Cause_to_amalgamate Parts:composto;
Filling Theme:olio di semi;Goal:stampini;
Placing Theme:composto;Goal:stampini;
Reshaping Patient:composto;Configuration:cestini;
Apply_heat Food:cestini;Heating_instrument:forno;
Placing Theme:cestini;Goal:vassoio;
Cause_to_be_included New_member:yogurt greco;Existing_member:cestini;

4) Crocchette di patate

Removing Source:patate;
Apply_heat Food:patate;
Removing Source:patate;
Grinding Patient:patate;
Separating Whole:uova;Parts:tuorli,albumi;
Cause_to_be_included New_member:sale,pepe;Existing_member:tuorli;
Cause_to_amalgamate Parts:tuorli;
Cause_to_be_included New_member:tuorli;Existing_member:patate;
Grinding Patient:noce moscata,formaggio;
Cause_to_be_included New_member:noce moscata,formaggio;Existing_member:patate;
Cause_to_amalgamate Parts:composto;
Cutting Item:composto;
Reshaping Patient:composto;Configuration:crocchette;
Placing Theme:uova;Goal:ciotola;

5) Pancackes

Apply_heat Food:burro;Container:pentola;
Separating Whole:uova;Parts:tuorli,albumi;
Placing Theme:tuorli;Goal:ciotola;
Cause_to_amalgamate Parts:tuorli;
Cause_to_be_included New_member:burro,latte;Existing_member:tuorli;
Cause_to_amalgamate Parts:tuorli;
Removing Source:farina,lievito;
Cause_to_be_included New_member:farina,lievito;Existing_member:composto;
Cause_to_amalgamate Parts:composto;
Cause_to_amalgamate Parts:zucchero,albumi;
Cause_to_be_included New_member:albumi;Existing_member:composto;
Cause_to_amalgamate Parts:composto;
Placing Theme:burro;Goal:padella;

6) Patate al forno

Removing Source:patate;
Removing Source:patate;
Cutting Item:patate;Pieces:cubetti;
Filling Theme:olio;Goal:teglia;
Apply_heat Food:acqua;Container:pentola;
Apply_heat Food:patate;Container:pentola;
Removing Source:patate;
Placing Theme:patate;Goal:ciotola;
Cause_to_be_included New_member:timo,olio,sale;Existing_member:patate;

7) Piadina Romagnola

Placing Theme:farina;Goal:ciotola;
Cause_to_be_included New_member:sale,strutto,bicarbonato,part#acqua;
  Existing_member:farina;
Cause_to_amalgamate Parts:composto;
Cause_to_be_included New_member:part#acqua;Existing_member:composto;
Cause_to_amalgamate Parts:composto;
Cause_to_be_included New_member:acqua;Existing_member:composto;
Cause_to_amalgamate Parts:composto;Whole:impasto;
Placing Theme:farina;Goal:piano di lavoro;

8) Pizzette

Placing Theme:acqua,lievito;Goal:ciotola;
Cause_to_amalgamate Parts:acqua,lievito;
Cause_to_be_included New_member:farina;Existing_member:composto;
Cause_to_be_included New_member:part#sale,part#olio,zucchero;Existing_member:composto;
Cause_to_amalgamate Parts:composto;Whole:impasto;
Storing Theme:impasto;Location:da parte;
Cutting Item:mozzarella;Pieces:cubetti;
Placing Theme:passata di pomodoro;Goal:ciotola;
Cause_to_be_included New_member:sale,pepe,origano,part#olio;
  Existing_member:passata di pomodoro;
Cause_to_amalgamate Parts:passata di pomodoro;
Storing Theme:passata di pomodoro;Location:da parte;
Filling Theme:farina;Goal:piano di lavoro;

9) Polpettine di tonno

Grinding Patient:parmigiano;
Grinding Patient:tonno;
Cause_to_amalgamate Parts:uova;
Placing Theme:pangrattato;Goal:ciotola;
Cause_to_be_included New_member:parmigiano,tonno,acciughe,uova;
  Existing_member:pangrattato;
Cause_to_amalgamate Parts:composto;
Cause_to_be_included New_member:sale,pepe;Existing_member:composto;
Cause_to_amalgamate Parts:composto;
Cutting Item:composto;Pieces:porzioni;
Reshaping Patient:porzioni;Configuration:polpettine;
Dunking Theme:polpettine;Substance:pangrattato;

10) Tiramisù

Separating Whole:uova;Parts:tuorli,albumi;
Cause_to_be_included New_member:part#zucchero;Existing_member:albumi;
Cause_to_amalgamate Parts:albumi;
Cause_to_be_included New_member:zucchero;Existing_member:tuorli;
Cause_to_amalgamate Parts:tuorli;
Cause_to_be_included New_member:caffè;Existing_member:tuorli;
Cause_to_amalgamate Parts:composto;
Cause_to_be_included New_member:albumi;Existing_member:composto;
Cause_to_amalgamate Parts:composto;

Placing Theme:part#composto;Goal:pirofila;
Dunking Theme:savoiardi;Substance:caffè;

11) Gnocchi

Placing Theme:patate,acqua;Goal:pentola;
Apply_heat Food:patate,acqua;Container:pentola;
Removing Source:patate;Theme:buccia;
Grinding Patient:patate;Grinder:schiacciapatate;
Cause_to_be_included New_member:uova,sale;Existing_member:patate;
Cause_to_be_included **New_member:farina;Existing_member:composto;**
Cause_to_amalgamate Parts:composto;
Placing Theme:farina;Goal:spianatoia;

12) Guacamole

Cutting Item:avocado;Pieces:a metà;
Removing Source:avocado;Theme:nocciolo;
Removing Source:avocado;Theme:buccia;
Cutting Item:avocado;Pieces:cubetti;
Grinding Patient:avocado;Grinder:forchetta;
Cause_to_be_included New_member:succo di lime,sale,pepe,olio;
Existing_member:avocado;
Cause_to_amalgamate Parts:composto;
Storing Theme:composto;
Cutting Item:scalogno;
Cutting Item:pomodori;Pieces:cubetti;
Removing Source:peperoncini verdi;Theme:semi;
Cutting Item:peperoncini verdi;Pieces:dadini;
Cause_to_be_included New_member:scalogno,pomodori,peperoncini verdi,olio;
Existing_member:composto;

13) Hamburger di ceci

Cutting Item:pancarrè;Pieces:cubetti;
Cutting Item:scalogno;
Removing Source:ceci;Theme:liquido di conservazione;
Placing Theme:ceci;Goal:mixer;
Cause_to_be_included New_member:pancarrè,uova,pangrattato,scalogno;
Existing_member:ceci;
Cause_to_amalgamate Parts:composto;
Grinding Patient:zenzero;
Cause_to_be_included New_member:zenzero,sale,pepe;Existing_member:composto;
Cause_to_amalgamate Parts:composto;
Placing Theme:composto;Goal:ciotola;
Cause_to_be_included New_member:pangrattato;Existing_member:composto;

14) Mousse al cioccolato

Grinding Patient:cioccolato fondente;Grinder:coltello;
Placing Theme:cioccolato fondente;Goal:ciotola;
Placing Theme:tuorli;Goal:tegame;
Cause_to_be_included New_member:miele,latte;Existing_member:tuorli;
Cause_to_amalgamate Parts:composto;
Apply_heat Food:composto;

Cause_to_be_included New_member:cioccolato fondente;Existing_member:composto;

Cause_to_amalgamate Parts:composto;

Cause_to_be_included **New_member:panna;Existing_member:composto;**

Cause_to_amalgamate Parts:composto;

Cause_to_be_included New_member:panna;Existing_member:composto;

Cause_to_amalgamate Parts:composto;

15) Plumcake

Cause_to_be_included New_member:semi baccello di vaniglia,burro,uova,sale,lievito, zucchero a velo,fecola;Existing_member:farina;

Cause_to_amalgamate Parts:semi baccello di vaniglia,burro,farina,uova,sale,lievito, zucchero a velo,fecola;Whole:composto;

Cause_to_be_included New_member:burro,farina;Existing_member:stampi;

16) Polpette di zucchine

Grinding Patient:zucchine;Grinder:grattugia a fori larghi;

Placing Theme:zucchine;Goal:colino;

Removing Source:zucchine;Theme:acqua di vegetazione;

Placing Theme:farina;Goal:ciotola;

Cause_to_be_included New_member:lievito;Existing_member:farina;

Cause_to_amalgamate Parts:composto;

Cause_to_be_included New_member:acqua;Existing_member:composto;

Cause_to_amalgamate Parts:composto;

Cause_to_be_included New_member:farina;Existing_member:composto;

17) Sformato di verdure

Cutting Item:melanzane;Pieces:fette;

Cutting Item:zucchine;Pieces:fette;

Removing Source:patate;Theme:buccia;

Cutting Item:patate;Pieces:fette;

Cutting Item:peperoni;Pieces:strisce;

Cutting Item:scamorza;Pieces:fette;

Cause_to_be_included New_member:part#parmigiano;Existing_member:pangrattato;

Cause_to_amalgamate Parts:composto;

Placing Theme:part#olio,part#parmigiano;Goal:pirofila;

Cause_to_amalgamate Parts:olio,parmigiano;

Cause_to_be_included New_member:melanzane,part#olio,part#sale,part#pepe;

Existing_member:composto;

Cause_to_be_included New_member:zucchine,part#olio,part#sale,part#pepe; Existing_member:composto;

Cause_to_be_included New_member:peperoni,part#olio,part#sale,part#pepe; Existing_member:composto;

Cause_to_be_included New_member:patate,part#olio,part#sale,part#pepe; Existing_member:composto;

Cause_to_be_included New_member:scamorza;Existing_member:composto;

Cause_to_be_included New_member:part#parmigiano;Existing_member:composto;

Cause_to_be_included New_member:pangrattato;Existing_member:composto;

18) Torta tenerina

Grinding Patient:cioccolato;Grinder:coltello;

Placing Goal:bastardella;Theme:cioccolato;

Apply_heat Food:cioccolato;

Cause_to_be_included New_member:part#burro;Existing_member:cioccolato;

Cause_to_amalgamate Parts:cioccolato;Whole:cioccolato;

Separating Parts:albumi,tuorli;Whole:uova;

Cause_to_be_included New_member:part#zucchero a velo;Existing_member:albumi;

Cause_to_amalgamate Parts:albumi;

Cause_to_be_included New_member:part#zucchero a velo;Existing_member:tuorli;

Cause_to_amalgamate Whole:tuorli;

Cause_to_be_included New_member:cioccolato;Existing_member:tuorli;

Cause_to_amalgamate Parts:cioccolato;

Cause_to_be_included New_member:uova;Existing_member:composto;

19) Zucchine alla scapece

Cutting Item:zucchine;Pieces:fettine;

Cause_to_be_included New_member:part#sale;Existing_member:zucchine;

Placing Theme:olio;Goal:ciotola;

Cause_to_be_included **New_member:menta,sale;Existing_member:olio;**

Cause_to_amalgamate Parts:composto;

Cutting Item:aglio;

Cause_to_be_included New_member:menta,aglio;Existing_member:composto;

20) Zuppa

Cutting Item:cipolla;

Cutting Item:zucca;Pieces:Fette;

Removing Source:zucca;Theme:buccia;

Cutting Item:zucca;Pieces:dadini;

Cutting Item:biete;Pieces:striscioline;

Placing Theme:olio;Goal:tegame;

Cause_to_be_included New_member:cipolla,ceci;Existing_member:olio;

Apply_heat Food:cipolla,bacche di ginepro,olio;

Cause_to_be_included New_member:ceci,biete,sale,pepe,alloro,acqua;Existing_member:cipolla, bacche di ginepro,olio;

Apply_heat Food:ceci,biete,sale,pepe,alloro,acqua,cipolla, bacche di ginepro,olio;

Removing Source:ceci,biete,sale,pepe,alloro,acqua,cipolla,bacche di ginepro,olio;Theme:alloro;

Appendix B

Chypher Queries

Pre-conditions

```
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO
// and the FRAME_ELEMENTs of type "Food" ENTITY nodes are ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT),
(e)-[r1:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND fe.name IN ['Food']

// If available, get the PERCEIVED_ENTITY nodes CREATED_FROM each PERCEIVED_ENTITY
// ENTITY nodes REFER_TO.
OPTIONAL MATCH (pe1)<-[r2:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)

// Compute the PERCEIVED_ELEMENTs quantity used by the last ACTION. 0 means "all the available quantity".
// If the available quantity is infinite, default to 1 to avoid Infinity - Infinity = NaN
WITH pe1, r2, a1,
CASE
    WHEN toFloat(apoc.text.regexGroups(e.value, r1.label + "\\[(\\d+)\\]")[0][1]) = 0 AND gds.util.isFinite(pe1.quantity) THEN pe1.quantity - SUM(r2.quantity)
    WHEN toFloat(apoc.text.regexGroups(e.value, r1.label + "\\[(\\d+)\\]")[0][1]) = 0 AND gds.util.isInfinite(pe1.quantity) THEN 1
    ELSE toFloat(apoc.text.regexGroups(e.value, r1.label + "\\[(\\d+)\\]")[0][1])
END AS newQuantity

// If the available quantity is more than 0 and subtracting the declared quantity is at least 0 the pre-condition is verified
WITH pe1.quantity - SUM(r2.quantity) - newQuantity >= 0 AND pe1.quantity - SUM(r2.quantity) > 0 AS Eval,
"Non ho abbastanza " + pe1.name + ". " AS NLExplanation, pe1, a1

// If the conflict is caused by a preceding ACTION, get the necessary data to build the HNPQ
// (ID of the conflicting ACTION, name of the conflicting FRAME, list of Ingredients involved in the conflicting ACTION)
OPTIONAL MATCH (pe1)<-[:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)-[:CREATED_BY]->(a2:ACTION)-[:REFERS_TO]->(:ENTITY)-[:REFERS_TO]->(pe3:PERCEIVED_ENTITY),
(a2:ACTION)-[:IS_A]->(:FRAME_INSTANCE)-[:INSTANCE_OF]->(f:FRAME) RETURN Eval,
COLLECT(ID(a2))[0] AS ConflictingAction, NLExplanation,
COLLECT(f.name)[0] AS ConflictingFrame,
apoc.text.join(COLLECT(DISTINCT pe3.name), ", ") AS OriginalEntity
```

Listing 1: Pre-conditions for Apply_heat

```
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO
// and the FRAME_ELEMENTs of type "Parts" and "Whole" ENTITY nodes are ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT),
(e)-[r1:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND fe.name IN ['Parts', 'Whole']

// If available, get the PERCEIVED_ENTITY nodes CREATED_FROM each PERCEIVED_ENTITY
// ENTITY nodes REFER_TO.
OPTIONAL MATCH (pe1)<-[r2:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)

// Compute the PERCEIVED_ELEMENTs quantity used by the last ACTION. 0 means "all the available
      quantity".
// If the available quantity is infinite, default to 1 to avoid Infinity - Infinity = NaN
WITH pe1, r2, a1,
CASE
  WHEN toFloat(apoc.text.regexGroups(e.value, r1.label + "\[(\d+)\]")[0][1]) = 0 AND gds.util.
     isFinite(pe1.quantity) THEN pe1.quantity - SUM(r2.quantity)
  WHEN toFloat(apoc.text.regexGroups(e.value, r1.label + "\[(\d+)\]")[0][1]) = 0 AND gds.util.
     isInfinite(pe1.quantity) THEN 1
  ELSE toFloat(apoc.text.regexGroups(e.value, r1.label + "\[(\d+)\]")[0][1])
END AS newQuantity

// If the available quantity is more than 0 and subtracting the declared quantity is at least
     0 the pre-condition is verified
WITH pe1.quantity - SUM(r2.quantity) - newQuantity >= 0 AND pe1.quantity - SUM(r2.quantity) >
     0 AS Eval,

// Builds the explanation concatenating "Non ho abbastanza" with the label of the insufficient
      PERCEIVED_ENTITY
"Non ho abbastanza " + pe1.name + ". " AS NLExplanation, pe1, a1

// If the conflict is caused by a preceding ACTION, get the necessary data to build the HNPQ
// (ID of the conflicting ACTION, name of the conflicting FRAME, list of Ingredients involved
     in the conflicting ACTION)
OPTIONAL MATCH (pe1)<-[:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)-[:CREATED_BY]->(a2:ACTION)-[:
     REFERS_TO]->(:ENTITY)-[:REFERS_TO]->(pe3:PERCEIVED_ENTITY),
(a2:ACTION)-[:IS_A]->(:FRAME_INSTANCE)-[:INSTANCE_OF]->(f:FRAME)
RETURN Eval, COLLECT(ID(a2))[0] AS ConflictingAction,
NLExplanation, COLLECT(f.name)[0] AS ConflictingFrame,
apoc.text.join(COLLECT(DISTINCT pe3.name), ", ") AS OriginalEntity
```

Listing 2: Pre-conditions for Cause_to_amalgamate

```
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO
// and the FRAME_ELEMENTs of type "New_member" and "Existing_member" ENTITY nodes are
     ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT),
(e)-[r1:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND fe.name IN ['New_member', 'Existing_member']

// If available, get the PERCEIVED_ENTITY nodes CREATED_FROM each PERCEIVED_ENTITY
OPTIONAL MATCH (pe1)<-[r2:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)

// Compute the PERCEIVED_ELEMENTs quantity used by the last ACTION. 0 means "all the available
      quantity".
// If the available quantity is infinite, default to 1 to avoid Infinity - Infinity = NaN
WITH pe1, r2, a1,
CASE
    WHEN toFloat(apoc.text.regexGroups(e.value, r1.label + "\\[(\\d+)\\]")[0][1]) = 0 AND gds.
    util.isFinite(pe1.quantity) THEN pe1.quantity - SUM(r2.quantity)
    WHEN toFloat(apoc.text.regexGroups(e.value, r1.label + "\\[(\\d+)\\]")[0][1]) = 0 AND gds.util.isInfinite(pe1.quantity) THEN 1
    ELSE toFloat(apoc.text.regexGroups(e.value, r1.label + "\\[(\\d+)\\]")[0][1])
END AS newQuantity

// If the available quantity is more than 0 and subtracting the declared quantity is at least 0 the pre-condition is verified
WITH pe1.quantity - SUM(r2.quantity) - newQuantity >= 0 AND pe1.quantity - SUM(r2.quantity) > 0 AS Eval,

// Builds the explanation concatenating "Non ho abbastanza" with the label of the insufficient PERCEIVED_ENTITY
"Non ho abbastanza " + pe1.name + ". " AS NLExplanation, pe1, a1

// If the conflict is caused by a preceding ACTION, get the necessary data to build the HNPQ
// (ID of the conflicting ACTION, name of the conflicting FRAME, list of Ingredients involved in the conflicting ACTION)
OPTIONAL MATCH (pe1)<-[:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)-[:CREATED_BY]->(a2:ACTION)-[:REFERS_TO]->(:ENTITY)-[:REFERS_TO]->(pe3:PERCEIVED_ENTITY),
(a2:ACTION)-[:IS_A]->(:FRAME_INSTANCE)-[:INSTANCE_OF]->(f:FRAME) RETURN Eval,
COLLECT(ID(a2))[0] AS ConflictingAction, NLExplanation,
COLLECT(f.name)[0] AS ConflictingFrame, apoc.text.join(COLLECT(DISTINCT pe3.name), ", ") AS OriginalEntity
```

Listing 3: Pre-conditions for Cause_to_be_included

```
//Condition 1: Verify that there is enough of the involved PERCEIVED_ELEMENTs to perform the ACTION
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO
// and the FRAME_ELEMENTs of type "Item" ENTITY node are ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT),
(e)-[r1:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND fe.name IN ['Item']

// If available, get the PERCEIVED_ENTITY nodes CREATED_FROM each PERCEIVED_ENTITY
// ENTITY nodes REFER_TO.
OPTIONAL MATCH (pe1)<-[r2:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)

// Compute the PERCEIVED_ELEMENTs quantity used by the last ACTION. 0 means "all the available quantity".
// If the available quantity is infinite, default to 1 to avoid Infinity - Infinity = NaN
WITH pe1, r2, a1,
CASE
  WHEN toFloat(apoc.text.regexGroups(e.value, r1.label + "\[(\d+)\]")[0][1]) = 0 AND gds.util.isFinite(pe1.quantity) THEN pe1.quantity - SUM(r2.quantity)
  WHEN toFloat(apoc.text.regexGroups(e.value, r1.label + "\[(\d+)\]")[0][1]) = 0 AND gds.util.isInfinite(pe1.quantity) THEN 1
  ELSE toFloat(apoc.text.regexGroups(e.value, r1.label + "\[(\d+)\]")[0][1])
END AS newQuantity

// If the available quantity is more than 0 and subtracting the declared quantity is at least 0 the pre-condition is verified
WITH pe1.quantity - SUM(r2.quantity) - newQuantity >= 0 AND pe1.quantity - SUM(r2.quantity) > 0 AS Eval,

// Builds the explanation concatenating "Non ho abbastanza" with the label of the insufficient PERCEIVED_ENTITY
```

```
"Non ho abbastanza " + pe1.name + ". " AS NLExplanation, pe1, a1

// If the conflict is caused by a preceding ACTION, get the necessary data to build the HNPQ
// (ID of the conflicting ACTION, name of the conflicting FRAME, list of Ingredients involved in the conflicting ACTION)
OPTIONAL MATCH (pe1)<-[:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)-[:CREATED_BY]->(a2:ACTION)-[:REFERS_TO]->(:ENTITY)-[:REFERS_TO]->(pe3:PERCEIVED_ENTITY),
(a2:ACTION)-[:IS_A]->(:FRAME_INSTANCE)-[:INSTANCE_OF]->(f:FRAME) RETURN Eval,
COLLECT(ID(a2))[0] AS ConflictingAction,
NLExplanation,
COLLECT(f.name)[0] AS ConflictingFrame,
apoc.text.join(COLLECT(DISTINCT pe3.name), ", ") AS OriginalEntity

//Condition 2: Verify that the involved PERCEIVED_ELEMENT is not a POWDER
UNION
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO and having the POWDER label
// and the FRAME_ELEMENTs of type "Item" ENTITY node are ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT {name: 'Item'}),
(e)-[:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND 'POWDER' IN labels(pe1)

// If at least one PERCEIVED_ELEMENT with the POWDER label is found, the pre-condition is not verified
WITH NOT COUNT(*) > 0 AS Eval

// If available, find a preceding version of the POWDER PERCEIVED_ELEMENT that did not have the POWDER label
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT {name: 'Item'}),
(e)-[:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->()
WITH Eval, pe1, a1
OPTIONAL MATCH (pe1)-[:CREATED_FROM*]->(pe2:PERCEIVED_ENTITY)<-[:REFERS_TO]-(:ENTITY)<-[:REFERS_TO]-(a2:ACTION)-[:IS_A]->(:FRAME_INSTANCE)-[:INSTANCE_OF]->(f:FRAME)
WHERE a1 <> a2 AND NOT 'POWDER' IN labels(pe2)

// Return the necessary information to build the HNPQ if a previous ACTION caused the PERCEIVED_ENTITY to acquire the POWDER label
RETURN Eval, COLLECT(ID(a2))[0] AS ConflictingAction,
pe1.name + '  in polvere.' AS NLExplanation,
COLLECT(f.name)[0] AS ConflictingFrame,
COLLECT(pe2.name)[0] AS OriginalEntity

//Condition 3: Verify that the involved PERCEIVED_ELEMENT is not a LIQUID
UNION
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO and having the LIQUID label
// and the FRAME_ELEMENTs of type "Item" ENTITY node are ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT {name: 'Item'}),
(e)-[:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND 'LIQUID' IN labels(pe1)

// If at least one PERCEIVED_ELEMENT with the LIQUID label is found, the pre-condition is not verified
WITH NOT COUNT(*) > 0 AS Eval

// If available, find a preceding version of the POWDER PERCEIVED_ELEMENT that did not have the LIQUID label
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT {name: 'Item'}),
(e)-[:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->()
WITH Eval, pe1, a1
OPTIONAL MATCH (pe1)-[:CREATED_FROM*]->(pe2:PERCEIVED_ENTITY)-[:CREATED_BY]->(a2:ACTION)-[:IS_A]->(:FRAME_INSTANCE)-[:INSTANCE_OF]->(f:FRAME)
WHERE a1 <> a2 AND NOT 'LIQUID' IN labels(pe2)

// Return the necessary information to build the HNPQ if a previous ACTION caused the PERCEIVED_ENTITY to acquire the POWDER label
RETURN Eval, COLLECT(ID(a2))[0] AS ConflictingAction, pe1.name + '  un liquido.' AS NLExplanation,
COLLECT(f.name)[0] AS ConflictingFrame, COLLECT(pe2.name)[0] AS OriginalEntity
```

Listing 4: Pre-conditions for Cutting

```
//Condition 1: Verify that there is enough of the involved PERCEIVED_ELEMENTs to perform the ACTION
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO
// and the FRAME_ELEMENTs of type "Patient" ENTITY node are ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT),
(e)-[r1:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND fe.name IN ['Patient']

// If available, get the PERCEIVED_ENTITY nodes CREATED_FROM each PERCEIVED_ENTITY
// ENTITY nodes REFER_TO.
OPTIONAL MATCH (pe1)<-[r2:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)

// Compute the PERCEIVED_ELEMENTs quantity used by the last ACTION. 0 means "all the available quantity".
// If the available quantity is infinite, default to 1 to avoid Infinity - Infinity = NaN
WITH pe1, r2, a1,
CASE
  WHEN toFloat(apoc.text.regexGroups(e.value, r1.label + "\[(\d+)\]")[0][1]) = 0 AND gds.util.isFinite(pe1.quantity) THEN pe1.quantity - SUM(r2.quantity)
  WHEN toFloat(apoc.text.regexGroups(e.value, r1.label + "\[(\d+)\]")[0][1]) = 0 AND gds.util.isInfinite(pe1.quantity) THEN 1
  ELSE toFloat(apoc.text.regexGroups(e.value, r1.label + "\[(\d+)\]")[0][1])
END AS newQuantity

// If the available quantity is more than 0 and subtracting the declared quantity is at least 0 the pre-condition is verified
WITH pe1.quantity - SUM(r2.quantity) - newQuantity >= 0 AND pe1.quantity - SUM(r2.quantity) > 0 AS Eval,

// Builds the explanation concatenating "Non ho abbastanza" with the label of the insufficient PERCEIVED_ENTITY
"Non ho abbastanza " + pe1.name + ". " AS NLExplanation, pe1, a1

// If the conflict is caused by a preceding ACTION, get the necessary data to build the HNPQ
// (ID of the conflicting ACTION, name of the conflicting FRAME, list of Ingredients involved in the conflicting ACTION)
OPTIONAL MATCH (pe1)<-[:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)-[:CREATED_BY]->(a2:ACTION)-[:REFERS_TO]->(:ENTITY)-[:REFERS_TO]->(pe3:PERCEIVED_ENTITY),
(a2:ACTION)-[:IS_A]->(:FRAME_INSTANCE)-[:INSTANCE_OF]->(f:FRAME) RETURN Eval,
COLLECT(ID(a2))[0] AS ConflictingAction,
NLExplanation,
COLLECT(f.name)[0] AS ConflictingFrame,
apoc.text.join(COLLECT(DISTINCT pe3.name), ", ") AS OriginalEntity
```

```

//Condition 2: Verify that the involved PERCEIVED_ELEMENT is not a POWDER
UNION
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO and having the POWDER label
// and the FRAME_ELEMENTs of type "Patient" ENTITY node are ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT {name: 'Patient'}),
(e)-[:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND 'POWDER' IN labels(pe1)

// If at least one PERCEIVED_ELEMENT with the POWDER label is found, the pre-condition is not verified
WITH NOT COUNT(*) > 0 AS Eval

// If available, find a preceding version of the POWDER PERCEIVED_ELEMENT that did not have the POWDER label
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT {name: 'Patient'}),
(e)-[:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->()
WITH Eval, pe1, a1
OPTIONAL MATCH (pe1)-[:CREATED_FROM*]->(pe2:PERCEIVED_ENTITY)<-[:REFERS_TO]-(:ENTITY)<-[:REFERS_TO]-(a2:ACTION)-[:IS_A]->(:FRAME_INSTANCE)-[:INSTANCE_OF]->(f:FRAME)
WHERE a1 <> a2 AND NOT 'POWDER' IN labels(pe2)

// Return the necessary information to build the HNPQ if a previous ACTION caused the PERCEIVED_ENTITY to acquire the POWDER label
RETURN Eval, COLLECT(ID(a2))[0] AS ConflictingAction,
pe1.name + '  in polvere.' AS NLExplanation,
COLLECT(f.name)[0] AS ConflictingFrame,
COLLECT(pe2.name)[0] AS OriginalEntity

//Condition 3: Verify that the involved PERCEIVED_ELEMENT is not a LIQUID
UNION
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO and having the LIQUID label
// and the FRAME_ELEMENTs of type "Patient" ENTITY node are ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT {name: 'Patient'}),
(e)-[:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND 'LIQUID' IN labels(pe1)

// If at least one PERCEIVED_ELEMENT with the LIQUID label is found, the pre-condition is not verified
WITH NOT COUNT(*) > 0 AS Eval

// If available, find a preceding version of the POWDER PERCEIVED_ELEMENT that did not have the LIQUID label
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT {name: 'Patient'}),
(e)-[:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->()
WITH Eval, pe1, a1
OPTIONAL MATCH (pe1)-[:CREATED_FROM*]->(pe2:PERCEIVED_ENTITY)-[:CREATED_BY]->(a2:ACTION)-[:IS_A]->(:FRAME_INSTANCE)-[:INSTANCE_OF]->(f:FRAME)
WHERE a1 <> a2 AND NOT 'LIQUID' IN labels(pe2)

// Return the necessary information to build the HNPQ if a previous ACTION caused the
```

```
PERCEIVED_ENTITY to acquire the LIQUID label
RETURN Eval, COLLECT(ID(a2))[0] AS ConflictingAction, pe1.name + '  un liquido.' AS NLExplanation,
COLLECT(f.name)[0] AS ConflictingFrame, COLLECT(pe2.name)[0] AS OriginalEntity
```

Listing 5: Pre-conditions for Grinding

```
//Condition 1: Verify that there is enough of the involved PERCEIVED_ELEMENT to perform the ACTION
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO
// and the FRAME_ELEMENTs of type "Theme" or "Substance" ENTITY node are ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT),
(e)-[r1:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND fe.name IN ['Theme', 'Substance]

// If available, get the PERCEIVED_ENTITY nodes CREATED_FROM each PERCEIVED_ENTITY
// ENTITY nodes REFER_TO.
OPTIONAL MATCH (pe1)<-[r2:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)

// Compute the PERCEIVED_ELEMENTs quantity used by the last ACTION. 0 means "all the available quantity".
// If the available quantity is infinite, default to 1 to avoid Infinity - Infinity = NaN
WITH pe1, r2, a1,
CASE toFloat(apoc.text.regexGroups(e.value, r1.label + "\[(\d+)\]")[0][1])
    WHEN 0 THEN pe1.quantity
    ELSE toFloat(apoc.text.regexGroups(e.value, r1.label + "\[(\d+)\]")[0][1])
END AS newQuantity

// If the available quantity is more than 0 and subtracting the declared quantity is at least 0 the pre-condition is verified
WITH pe1.quantity - SUM(r2.quantity) - newQuantity >= 0 AS Eval,

// Builds the explanation concatenating "Non ho abbastanza" with the label of the insufficient PERCEIVED_ENTITY
"Non ho abbastanza " + pe1.name + ". " AS NLExplanation, pe1, a1

// If the conflict is caused by a preceding ACTION, get the necessary data to build the HNPQ
// (ID of the conflicting ACTION, name of the conflicting FRAME, list of Ingredients involved in the conflicting ACTION)
OPTIONAL MATCH (pe1)<-[:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)-[:CREATED_BY]->(a2:ACTION)-[:REFERS_TO]->(:ENTITY)-[:REFERS_TO]->(pe3:PERCEIVED_ENTITY),
(a2:ACTION)-[:IS_A]->(:FRAME_INSTANCE)-[:INSTANCE_OF]->(f:FRAME)
RETURN Eval, COLLECT(ID(a2))[0] AS ConflictingAction, NLExplanation,
COLLECT(f.name)[0] AS ConflictingFrame, apoc.text.join(COLLECT(DISTINCT pe3.name), ", ") AS OriginalEntity

//Condition 2: Verify that the involved PERCEIVED_ELEMENT in "Theme" is not a POWDER
UNION

// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO and having the POWDER label
// and the FRAME_ELEMENTs of type "Theme" ENTITY node are ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT {name: 'Theme'}),
(e)-[:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND 'POWDER' IN labels(pe1)

// If at least one PERCEIVED_ELEMENT with the POWDER label is found, the pre-condition is not verified
```

```
WITH NOT COUNT(*) > 0 AS Eval

// If available, find a preceding version of the POWDER PERCEIVED_ELEMENT that did not have the POWDER label
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT {name: 'Theme'}),
(e)-[:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() WITH Eval, pe1, a1
OPTIONAL MATCH (pe1)-[:CREATED_FROM*]->(pe2:PERCEIVED_ENTITY)<-[:REFERS_TO]-(:ENTITY)<-[:REFERS_TO]-(a2:ACTION)-[:IS_A]->(:FRAME_INSTANCE)-[:INSTANCE_OF]->(f:FRAME)
WHERE a1 <> a2 AND NOT 'POWDER' IN labels(pe2)

// Return the necessary information to build the HNPQ if a previous ACTION caused the PERCEIVED_ENTITY to acquire the POWDER label
RETURN Eval, COLLECT(ID(a2))[0] AS ConflictingAction,
pe1.name + '  in polvere.' AS NLExplanation, COLLECT(f.name)[0] AS ConflictingFrame, COLLECT(pe2.name)[0] AS OriginalEntity

// Condition 3: Verify that the involved PERCEIVED_ELEMENT in "Theme" is not a LIQUID
UNION

// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO and having the LIQUID label
// and the FRAME_ELEMENTs of type "Theme" ENTITY node are ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT {name: 'Theme'}),
(e)-[:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND 'LIQUID' IN labels(pe1)

// If at least one PERCEIVED_ELEMENT with the LIQUID label is found, the pre-condition is not verified
WITH NOT COUNT(*) > 0 AS Eval

// If available, find a preceding version of the LIQUID PERCEIVED_ELEMENT that did not have the LIQUID label
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT {name: 'Theme'}),
(e)-[:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() WITH Eval, pe1, a1
OPTIONAL MATCH (pe1)-[:CREATED_FROM*]->(pe2:PERCEIVED_ENTITY)-[:CREATED_BY]->(a2:ACTION)-[:IS_A]->(:FRAME_INSTANCE)-[:INSTANCE_OF]->(f:FRAME)
WHERE a1 <> a2 AND NOT 'LIQUID' IN labels(pe2)

// Return the necessary information to build the HNPQ if a previous ACTION caused the PERCEIVED_ENTITY to acquire the LIQUID label
RETURN Eval, COLLECT(ID(a2))[0] AS ConflictingAction,
pe1.name + '  un liquido.' AS NLExplanation, COLLECT(f.name)[0] AS ConflictingFrame, COLLECT(pe2.name)[0] AS OriginalEntity

// Condition 4: Verify that the PERCEIVED_ELEMENT in "Substance" is either a LIQUID or a POWDER
UNION

// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO and having the LIQUID label
// and the FRAME_ELEMENTs of type "Substance" ENTITY node are ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT {name: 'Substance'}),
(e)-[:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND ('LIQUID' IN labels(pe1) OR 'POWDER' IN labels(pe1))
WITH COUNT(*) > 0 AS Eval

```

```
// If available, find a preceding version of the PERCEIVED_ELEMENT that had the LIQUID or the POWDER label
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT {name: 'Substance'}),
(e)-[:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() WITH Eval, pe1, a1
OPTIONAL MATCH (pe1)-[:CREATED_FROM*]->(pe2:PERCEIVED_ENTITY)-[:CREATED_BY]->(a2:ACTION)-[:IS_A]->(:FRAME_INSTANCE)-[:INSTANCE_OF]->(f:FRAME)
WHERE a1 <> a2 AND ('LIQUID' IN labels(pe2) OR 'POWDER' IN labels(pe2))

// Return the explanation. Since no ACTIONS can remove the labels POWDER or LIQUID, there is no check over possible conflicting ACTIONs
RETURN Eval, COLLECT(ID(a2))[0] AS ConflictingAction,
pe1.name + ' non  un liquido o una polvere.' AS NLExplanation, COLLECT(f.name)[0] AS ConflictingFrame, COLLECT(pe2.name)[0] AS OriginalEntity
```

Listing 6: Pre-conditions for Dunking

```
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO
// and the FRAME_ELEMENTs of type "Theme" ENTITY nodes are ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT),
(e)-[r1:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND fe.name IN ['Theme']

// If available, get the PERCEIVED_ENTITY nodes CREATED_FROM each PERCEIVED_ENTITY
OPTIONAL MATCH (pe1)<-[r2:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)

// Compute the PERCEIVED_ELEMENTs quantity used by the last ACTION. 0 means "all the available quantity".
// If the available quantity is infinite, default to 1 to avoid Infinity - Infinity = NaN
WITH pe1, r2, a1,
CASE
WHEN toFloat(apoc.text.regexGroups(e.value, r1.label + "\\[(\\d+)\\]")[0][1]) = 0 AND gds.util.isFinite(pe1.quantity) THEN pe1.quantity - SUM(r2.quantity)
WHEN toFloat(apoc.text.regexGroups(e.value, r1.label + "\\[(\\d+)\\]")[0][1]) = 0 AND gds.util.isInfinite(pe1.quantity) THEN 1
ELSE toFloat(apoc.text.regexGroups(e.value, r1.label + "\\[(\\d+)\\]")[0][1])
END AS newQuantity

// If the available quantity is more than 0 and subtracting the declared quantity is at least 0 the pre-condition is verified
WITH pe1.quantity - SUM(r2.quantity) - newQuantity >= 0 AND pe1.quantity - SUM(r2.quantity) > 0 AS Eval,

// Builds the explanation concatenating "Non ho abbastanza" with the label of the insufficient PERCEIVED_ENTITY
"Non ho abbastanza " + pe1.name + ". " AS NLExplanation, pe1, a1

// If the conflict is caused by a preceding ACTION, get the necessary data to build the HNPQ
// (ID of the conflicting ACTION, name of the conflicting FRAME, list of Ingredients involved in the conflicting ACTION)
OPTIONAL MATCH (pe1)<-[:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)-[:CREATED_BY]->(a2:ACTION)-[:REFERS_TO]->(:ENTITY)-[:REFERS_TO]->(pe3:PERCEIVED_ENTITY),
(a2:ACTION)-[:IS_A]->(:FRAME_INSTANCE)-[:INSTANCE_OF]->(f:FRAME)
RETURN Eval, COLLECT(ID(a2))[0] AS ConflictingAction, NLExplanation,
COLLECT(f.name)[0] AS ConflictingFrame, apoc.text.join(COLLECT(DISTINCT pe3.name), ", ") AS OriginalEntity
```

Listing 7: Pre-conditions for Placing

```
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO
// and the FRAME_ELEMENTs of type "Patient" ENTITY nodes are ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT),
(e)-[r1:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND fe.name IN ['Patient']

// If available, get the PERCEIVED_ENTITY nodes CREATED_FROM each PERCEIVED_ENTITY
OPTIONAL MATCH (pe1)<-[r2:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)

// Compute the PERCEIVED_ELEMENTs quantity used by the last ACTION. 0 means "all the available quantity".
// If the available quantity is infinite, default to 1 to avoid Infinity - Infinity = NaN
WITH pe1, r2, a1,
CASE
WHEN toFloat(apoc.text.regexGroups(e.value, r1.label + "\\[(\\d+)\\]")[0][1]) = 0 AND gds.util.isFinite(pe1.quantity) THEN pe1.quantity - SUM(r2.quantity)
WHEN toFloat(apoc.text.regexGroups(e.value, r1.label + "\\[(\\d+)\\]")[0][1]) = 0 AND gds.util.isInfinite(pe1.quantity) THEN 1
ELSE toFloat(apoc.text.regexGroups(e.value, r1.label + "\\[(\\d+)\\]")[0][1])
END AS newQuantity

// If the available quantity is more than 0 and subtracting the declared quantity is at least 0 the pre-condition is verified
WITH pe1.quantity - SUM(r2.quantity) - newQuantity >= 0 AND pe1.quantity - SUM(r2.quantity) > 0 AS Eval,

// Builds the explanation concatenating "Non ho abbastanza" with the label of the insufficient PERCEIVED_ENTITY
"Non ho abbastanza " + pe1.name + ". " AS NLExplanation, pe1, a1

// If the conflict is caused by a preceding ACTION, get the necessary data to build the HNPQ
// (ID of the conflicting ACTION, name of the conflicting FRAME, list of Ingredients involved in the conflicting ACTION)
OPTIONAL MATCH (pe1)<-[:CREATED_FROM]-(pe2:PERCEIVED_ENTITY)-[:CREATED_BY]->(a2:ACTION)-[:REFERS_TO]->(:ENTITY)-[:REFERS_TO]->(pe3:PERCEIVED_ENTITY),
(a2:ACTION)-[:IS_A]->(:FRAME_INSTANCE)-[:INSTANCE_OF]->(f:FRAME)
RETURN Eval, COLLECT(ID(a2))[0] AS ConflictingAction, NLExplanation,
COLLECT(f.name)[0] AS ConflictingFrame, apoc.text.join(COLLECT(DISTINCT pe3.name), ", ") AS OriginalEntity
```

Listing 8: Pre-conditions for Reshaping

```
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO
// and the FRAME_ELEMENTs of type "Parts" or "Whole" ENTITY nodes are ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT),

// Get the mix PERCEIVED_ELEMENT the target ENTITY nodes may be part of
(e)-[r1:REFERS_TO]->(pe1:PERCEIVED_ENTITY)<-[:CREATED_FROM]-(pe2:PERCEIVED_ENTITY {name: 'Composto'}),

// Get all the ENTITY nodes the ACTION REFERS_TO
(a1)-[:REFERS_TO]->(e2:ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND fe.name IN ['Parts', 'Whole']

// If the linked entities are found in the same mix or if none of the entities is found in the mix, the precondition is verified
// This covers both the case of separating two previously mixed ingredients and the case in
```

```
    which a part of an ingredient is removed from it
WITH COUNT(DISTINCT pe1) = COUNT(DISTINCT e2) AS inMix
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT),
(e)-[r1:REFERS_TO]->(pe1:PERCEIVED_ENTITY)<-[:CREATED_FROM]-(pe2:PERCEIVED_ENTITY {name: '
    Composto'}),
(a1)-[:REFERS_TO]->(e2:ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND fe.name IN ['Parts', 'Whole']
RETURN COUNT(*) = 0 OR inMix AS Eval, null AS ConflictingFrame, null AS ConflictingAction,
"Gli ingredienti non sono uniti." AS NLExplanation, null AS OriginalEntity
```

Listing 9: Pre-conditions for Separating

Post-conditions

```
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO,
    the DIALOGUE_DOMAIN
// and the FRAME_ELEMENTs of type "Food" ENTITY node are ASSIGNED_TO.
MATCH (a:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(:FRAME_ELEMENT {name: 'Food'}),
(pe:PERCEIVED_ENTITY)<-[r2:REFERS_TO]-(e),
(d:DIALOGUE_DOMAIN {name: 'Cooking'})
WHERE NOT (a)-[:IS_FOLLOWED_BY]->()

// Get how much of the target PERCEIVED_ENTITY has been used to produce PERCEIVED_ENTITY nodes
OPTIONAL MATCH (pe)<-[r3:CREATED_FROM]-(:PERCEIVED_ENTITY)

// Create a new PERCEIVED_ENTITY having the same name of the PERCEIVED_ENTITY nodes the action
     is being applied to.
// Specify that the new node is CREATED_FROM the PERCEIVED_ENTITY the ENTITY REFERS_TO, that
    it was CREATED_BY
// the last ACTION and that it BELONGS_TO the DIALOGUE_DOMAIN
CREATE (peNew:PERCEIVED_ENTITY {name: pe.name})
CREATE (peNew)-[r1:CREATED_FROM]->(pe)
CREATE (peNew)-[:CREATED_BY]->(a)
CREATE (peNew)-[:BELONGS_TO]->(d)

// Put a quantity property on the new CREATED_FROM relationship specifying how much of the
    PERCEIVED_ENTITIES was used.
// This is also the available quantity of the new PERCEIVED_ENTITY nodes.
WITH e, r2.label AS label, toFloat(apoc.text.regexGroups(e.value, r2.label + "\[(\d+)\]")
    [0][1]) AS quantity, pe, r1, r2, peNew
WITH peNew, r1,
CASE
WHEN quantity > 0 THEN quantity
WHEN quantity = 0 AND gds.util.isFinite(pe.quantity) THEN pe.quantity - SUM(r3.quantity)
WHEN quantity = 0 AND gds.util.isInfinite(pe.quantity) THEN 1.0
END AS usedValue, e, r2, pe
SET r1.quantity = usedValue
SET peNew.quantity = usedValue

// If the PERCEIVED_ENTITY involved in the action was MELTABLE, the new PERCEIVED_ENTITY is
    also a LIQUID
FOREACH (i IN CASE WHEN 'MELTABLE' IN labels(pe) THEN [1] ELSE [] END | SET peNew:LIQUID)

// Remove quantity information from the ENTITY nodes
WITH e, COLLECT(r2.label) AS labels
SET e.value= apoc.text.join(labels, ',')
```

Listing 10: Post-conditions for Apply_heat

```
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO, the DIALOGUE_DOMAIN
// and the FRAME_ELEMENTs of type "Parts" ENTITY node are ASSIGNED_TO.
MATCH (a:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(:FRAME_ELEMENT {name: 'Parts'}),
(pe:PERCEIVED_ENTITY)<-[r2:REFERS_TO]-(e),
(d:DIALOGUE_DOMAIN {name: 'Cooking'})
WHERE NOT(a)-[:IS_FOLLOWED_BY]->()

// Get how much of the target PERCEIVED_ENTITY has been used to produce PERCEIVED_ENTITY nodes
OPTIONAL MATCH (pe)<-[r3:CREATED_FROM]-(:PERCEIVED_ENTITY)

// Create a new PERCEIVED_ENTITY named "Composto".
// Specify that the new node is CREATED_FROM the PERCEIVED_ENTITY the ENTITY REFERS_TO, that it was CREATED_BY
// the last ACTION and that it BELONGS_TO the DIALOGUE_DOMAIN
MERGE (peNew:PERCEIVED_ENTITY {name: 'Composto', wholeId: ID(a)})
MERGE (peNew)-[r1:CREATED_FROM]->(pe)
MERGE (peNew)-[:CREATED_BY]->(a)
MERGE (peNew)-[:BELONGS_TO]->(d)

// Put a quantity property on the new CREATED_FROM relationship specifying how much of the PERCEIVED_ENTITIES was used.
// This is also the available quantity of the new PERCEIVED_ENTITY nodes
WITH e, toFloat(apoc.text.regexGroups(e.value, r2.label + "\[(\d+)\]")[0][1]) AS quantity, pe, r1, r2, r3
WITH r1,
CASE
    WHEN quantity > 0 THEN quantity
    WHEN quantity = 0 AND gds.util.isFinite(pe.quantity) THEN pe.quantity - SUM(r3.quantity)
    WHEN quantity = 0 AND gds.util.isInfinite(pe.quantity) THEN 1.0
END AS usedValue, e, r2
SET r1.quantity = usedValue

// Remove quantity information from the ENTITY nodes
WITH e, COLLECT(r2.label) AS labels
SET e.value= apoc.text.join(labels, ',')
```

Listing 11: Post-conditions for Cause_to_amalgamate

```
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO, the DIALOGUE_DOMAIN
// and the FRAME_ELEMENTs of type "Patient" ENTITY node are ASSIGNED_TO.
MATCH (a:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(:FRAME_ELEMENT {name: 'Patient'}),
(pe:PERCEIVED_ENTITY)<-[r2:REFERS_TO]-(e),
(d:DIALOGUE_DOMAIN {name: 'Cooking'})
WHERE NOT (a)-[:IS_FOLLOWED_BY]->()

// Create a new PERCEIVED_ENTITY having the same name of the PERCEIVED_ENTITY nodes the action is being applied to.
// Specify that the new nodee is CREATED_FROM the PERCEIVED_ENTITY the ENTITY REFERS_TO, that it was CREATED_BY
// the last ACTION and that it BELONGS_TO the DIALOGUE_DOMAIN
CREATE (peNew:PERCEIVED_ENTITY:POWDER {name: pe.name})
CREATE (peNew)-[r1:CREATED_FROM]->(pe)
CREATE (peNew)-[:CREATED_BY]->(a)
CREATE (peNew)-[:BELONGS_TO]->(d)

```

```
// Put a quantity property on the new CREATED_FROM relationship specifying how much of the
     PERCEIVED_ENTITIES was used.
// This is also the available quantity of the new PERCEIVED_ENTITY nodes.
WITH e, r2.label AS label, toFloat(apoc.text.regexGroups(e.value, r2.label + "\[(\d+)\]")
     [0][1]) AS quantity, pe, r1, r2, peNew
WITH peNew, r1,
CASE
    WHEN quantity > 0 THEN quantity
    WHEN quantity = 0 AND gds.util.isFinite(pe.quantity) THEN pe.quantity
    WHEN quantity = 0 AND gds.util.isInfinite(pe.quantity) THEN 1.0
END AS usedValue, e, r2
SET r1.quantity = usedValue
SET peNew.quantity = usedValue

// Remove quantity information from the ENTITY nodes
WITH e, COLLECT(r2.label) AS labels
SET e.value= apoc.text.join(labels, ',')
```

Listing 12: Post-conditions for Grinding

```
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO,
     the DIALOGUE_DOMAIN
// and the FRAME_ELEMENTs of type "Theme" and "Substance" ENTITY nodes are ASSIGNED_TO.
MATCH (a:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT),
(pe:PERCEIVED_ENTITY)<-[r2:REFERS_TO]-(e),
(d:DIALOGUE_DOMAIN {name: 'Cooking'})
WHERE NOT(a)-[:IS_FOLLOWED_BY]->() AND fe.name IN ['Theme', 'Substance']

// Get how much of the target PERCEIVED_ENTITY has been used to produce PERCEIVED_ENTITY nodes
OPTIONAL MATCH (pe)<-[r3:CREATED_FROM]-(:PERCEIVED_ENTITY)

// Create a new PERCEIVED_ENTITY named "Composto".
// Specify that the new node is CREATED_FROM the PERCEIVED_ENTITY the ENTITY REFERS_TO, that
     it was CREATED_BY
// the last ACTION and that it BELONGS_TO the DIALOGUE_DOMAIN
MERGE (peNew:PERCEIVED_ENTITY {name: 'Composto', wholeId: ID(a)})
MERGE (peNew)-[r1:CREATED_FROM]->(pe)
MERGE (peNew)-[:CREATED_BY]->(a)
MERGE (peNew)-[:BELONGS_TO]->(d)

// Put a quantity property on the new CREATED_FROM relationship specifying how much of the
     PERCEIVED_ENTITIES was used.
// This is also the available quantity of the new PERCEIVED_ENTITY nodes
WITH e, toFloat(apoc.text.regexGroups(e.value, r2.label + "\[(\d+)\]")[0][1]) AS quantity, pe,
      r1, r2, r3
WITH r1,
CASE
    WHEN quantity > 0 THEN quantity
    WHEN quantity = 0 AND gds.util.isFinite(pe.quantity) THEN pe.quantity - SUM(r3.quantity)
    WHEN quantity = 0 AND gds.util.isInfinite(pe.quantity) THEN 1.0
END AS usedValue, e, r2
SET r1.quantity = usedValue

// Remove quantity information from the ENTITY nodes
WITH e, COLLECT(r2.label) AS labels
SET e.value= apoc.text.join(labels, ',')
```

Listing 13: Post-conditions for Dunking

```
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO, the DIALOGUE_DOMAIN
// and the FRAME_ELEMENTs of type "Theme" ENTITY node are ASSIGNED_TO.
MATCH (a:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(:FRAME_ELEMENT {name: 'Theme'}),
(pe:PERCEIVED_ENTITY)<-[r2:REFERS_TO]-(e),
(a)-[:REFERS_TO]->(e2:ENTITY)-[:ASSIGNED_TO]->(:FRAME_ELEMENT {name: 'Area'})
(d:DIALOGUE_DOMAIN {name: 'Cooking'})
WHERE NOT (a)-[:IS_FOLLOWED_BY]->()

// Create a new PERCEIVED_ENTITY having the same name of the PERCEIVED_ENTITY nodes the action is being applied to and
// a PERCEIVED_ENTITY representing the place where to put the "Theme" PERCEIVED_ENTITY
CREATE (pePlace:PERCEIVED_ENTITY {name: e2.value})
CREATE (peNew:PERCEIVED_ENTITY {name: pe.name})
CREATE (peNew)-[:PLACED_IN]->(pePlace)
CREATE (peNew)-[r1:CREATED_FROM]->(pe)
CREATE (peNew)-[:CREATED_BY]->(a)
CREATE (peNew)-[:BELONGS_TO]->(d)

// Put a quantity property on the new CREATED_FROM relationship specifying how much of the PERCEIVED_ENTITIEs was used.
// This is also the available quantity of the new PERCEIVED_ENTITY nodes.
WITH e, r2.label AS label, toFloat(apoc.text.regexGroups(e.value, r2.label + "\[(\d+)\]")[0][1]) AS quantity, pe, r1, r2, peNew
WITH peNew, r1,
CASE
WHEN quantity > 0 THEN quantity
WHEN quantity = 0 AND gds.util.isFinite(pe.quantity) THEN pe.quantity
WHEN quantity = 0 AND gds.util.isInfinite(pe.quantity) THEN 1.0
END AS usedValue, e, r2
SET r1.quantity = usedValue
SET peNew.quantity = usedValue

// Remove quantity information from the ENTITY nodes
WITH e, COLLECT(r2.label) AS labels
SET e.value= apoc.text.join(labels, ',')
```

Listing 14: Post-conditions for Placing

```
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO, the DIALOGUE_DOMAIN
// and the FRAME_ELEMENTs of type "Patient" ENTITY node are ASSIGNED_TO.
MATCH (a:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(:FRAME_ELEMENT {name: 'Theme'}),
(pe:PERCEIVED_ENTITY)<-[r2:REFERS_TO]-(e),
(a)-[:REFERS_TO]->(e2:ENTITY)-[:ASSIGNED_TO]->(:FRAME_ELEMENT {name: 'Source'}),
(e2)-[:REFERS_TO]->(pe2:PERCEIVED_ENTITY)<-[:PLACED_IN]-(pe)
(d:DIALOGUE_DOMAIN {name: 'Cooking'})
WHERE NOT (a)-[:IS_FOLLOWED_BY]->()

// Create a new PERCEIVED_ENTITY having the same name of the PERCEIVED_ENTITY nodes the action is being applied to.
CREATE (peNew:PERCEIVED_ENTITY {name: pe.name})
CREATE (peNew)-[r1:CREATED_FROM]->(pe)
CREATE (peNew)-[:CREATED_BY]->(a)
CREATE (peNew)-[:BELONGS_TO]->(d)

// Put a quantity property on the new CREATED_FROM relationship specifying how much of the
```

```
      PERCEIVED_ENTITIEs was used.
// This is also the available quantity of the new PERCEIVED_ENTITY nodes.
WITH e, r2.label AS label, toFloat(apoc.text.regexGroups(e.value, r2.label + "\[(\d+)\]")
      [0][1]) AS quantity, pe, r1, r2, peNew
WITH peNew, r1,
CASE
WHEN quantity > 0 THEN quantity
WHEN quantity = 0 AND gds.util.isFinite(pe.quantity) THEN pe.quantity
WHEN quantity = 0 AND gds.util.isInfinite(pe.quantity) THEN 1.0
END AS usedValue, e, r2
SET r1.quantity = usedValue
SET peNew.quantity = usedValue

// Remove quantity information from the ENTITY nodes
WITH e, COLLECT(r2.label) AS labels
SET e.value= apoc.text.join(labels, ',')
```

Listing 15: Post-conditions for Removing

```
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO,
      the DIALOGUE_DOMAIN
// and the FRAME_ELEMENTs of type "Patient" ENTITY node are ASSIGNED_TO.
MATCH (a:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(:FRAME_ELEMENT {name: 'Patient'}),
(pe:PERCEIVED_ENTITY)<-[r2:REFERS_TO]-(e),
(a)-[:REFERS_TO]->(e2:ENTITY)-[:ASSIGNED_TO]->(:FRAME_ELEMENT {name: 'Result'})
(d:DIALOGUE_DOMAIN {name: 'Cooking'})
WHERE NOT (a)-[:IS_FOLLOWED_BY]->()

// Create a new PERCEIVED_ENTITY having the name of the "Result" entity.
CREATE (peNew:PERCEIVED_ENTITY {name: e2.value})
CREATE (peNew)-[r1:CREATED_FROM]->(pe)
CREATE (peNew)-[:CREATED_BY]->(a)
CREATE (peNew)-[:BELONGS_TO]->(d)

// Put a quantity property on the new CREATED_FROM relationship specifying how much of the
      PERCEIVED_ENTITIEs was used.
// This is also the available quantity of the new PERCEIVED_ENTITY nodes.
WITH e, r2.label AS label, toFloat(apoc.text.regexGroups(e.value, r2.label + "\[(\d+)\]")
      [0][1]) AS quantity, pe, r1, r2, peNew
WITH peNew, r1,
CASE
WHEN quantity > 0 THEN quantity
WHEN quantity = 0 AND gds.util.isFinite(pe.quantity) THEN pe.quantity
WHEN quantity = 0 AND gds.util.isInfinite(pe.quantity) THEN 1.0
END AS usedValue, e, r2
SET r1.quantity = usedValue
SET peNew.quantity = usedValue

// Remove quantity information from the ENTITY nodes
WITH e, COLLECT(r2.label) AS labels
SET e.value= apoc.text.join(labels, ',')
```

Listing 16: Post-conditions for Reshaping

```
// Get the last ACTION, the ENTITY nodes it refers to, the PERCEIVED_ELEMENTs they REFER_TO,
      the DIALOGUE_DOMAIN
// and the FRAME_ELEMENTs of type "Patient" ENTITY node are ASSIGNED_TO.
MATCH (a1:ACTION)-[:REFERS_TO]->(e:ENTITY)-[:ASSIGNED_TO]->(fe:FRAME_ELEMENT),
(e)-[r1:REFERS_TO]->(pe1:PERCEIVED_ENTITY)
WHERE NOT (a1)-[:IS_FOLLOWED_BY]->() AND fe.name IN ['Parts', 'Whole']

// If the target PERCEIVED_ENTITY nodes are part of a mix, also get the mix
OPTIONAL MATCH (pe1)<-[:CREATED_FROM]-(pe2:PERCEIVED_ENTITY {name: 'Composto'})

// If the target PERCEIVED_ENTITY nodes are part of a mix, the resulting quantity from the separation
// is the same as before they became part of the mix
WITH pe1, pe2, e,
CASE
WHEN pe2 IS NOT null THEN 1
ELSE pe2.quantity
END AS quantity,

// If the target PERCEIVED_ENTITY nodes are part of a mix, the newly created PERCEIVED_ENTITY nodes
// should be CREATED_FROM the mix. They should be created from the ingredients themselves otherwise
CASE
WHEN pe2 IS NOT null THEN pe2
ELSE pe1
END AS targetPE

// Create and link the new PERCEIVED_ENTITY nodes
CREATE (res:PERCEIVED_ENTITY {quantity: quantity})
MERGE (res)-[:CREATED_FROM]->(targetPE)
```

Listing 17: Post-conditions for Separating

Bibliography

[1] Gabriella Airenti, Bruno Giuseppe Bara, and Marco Colombetti. “Conversation and behavior games in the pragmatics of dialogue”. In: *Cognitive Science* 17.2 (1993), pp. 197–256.

[2] Jens Allwood. “A framework for studying human multimodal communication”. In: *Coverbal synchrony in human-machine interaction* 17 (2013).

[3] Jens Allwood. “An Activity Based Approach To Pragmatics”. In: John Benjamins, 1995.

[4] Jens Allwood, Joakim Nivre, and Elisabeth Ahlsén. “On the semantics and pragmatics of linguistic feedback”. In: *Journal of Semantics* 9 (1992), pp. 1–26.

[5] Egbert Ammicht, Eric Fosler-Lussier, and Alexandros Potamianos. *System and method for representing and resolving ambiguity in spoken dialogue systems.* US Patent App. 10/170,510. Google Patents, 2003.

[6] Scott AnderBois. *Issues and alternatives.* University of California, Santa Cruz, 2011.

[7] Nicholas Asher and Brian Reese. “Negative bias in polar questions”. In: *Proceedings of Sinn und Bedeutung.* Vol. 9. 2005, pp. 30–43.

[8] Francesco Avolio. “Ma nuje comme parlamme? Problemi di descrizione e classificazione dello spazio dialettale" campano"”. In: *Romance Philology* 54.1 (2000), pp. 1–28.

[9] 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, Volume 1.* 1998, pp. 86–90.

[10] Bruno Giuseppe Bara. *Cognitive pragmatics.* Cambridge, MA: MIT Press, 2010.

[11] Bruno Giuseppe Bara. “Pragmatica cognitiva: i processi mentali della comunicazione”. In: (1999).

[12] Simon Baron-Cohen. “Theory of mind in normal development and autism”. In: *Prisme* 34.1 (2001), pp. 74–183.

[13] Carla Bazzanella. *Le facce del parlare. Un approccio pragmatico all’italiano parlato.* Vol. 17. LA NUOVA ITALIACollana: Biblioteca di Italiano e oltre, 1994.

[14] Carla Bazzanella. “Linguistica e pragmatica del linguaggio. Un’introduzione.” In: (2005).

[15] Tilman Becker, Nate Blaylock, Ciprian Gerstenberger, Ivana Kruijff-Korbayová, Andreas Korthauer, Manfred Pinkal, Michael Pitz, Peter Poller, and Jan Schehl. "Natural and intuitive multimodal dialogue for in-car applications: The SAMMIE system". In: *Frontiers in Artificial Intelligence and Applications* 141 (2006), p. 612.

[16] Robbert-Jan Beun and Rogier M van Eijk. "Conceptual discrepancies and feedback in human-computer interaction". In: *Proceedings of the conference on Dutch directions in HCI.* 2004, p. 13.

[17] Elizabeth Black and Katie Atkinson. "Agreeing what to do". In: *International Workshop on Argumentation in Multi-Agent Systems.* Springer. 2010, pp. 12–30.

[18] Ned Block. "On a confusion about a function of consciousness". In: *Behavioral and brain sciences* 18.2 (1995), pp. 227–247.

[19] Paul Boersma and David J. M. Weenink. "Praat, a system for doing phonetics by computer". In: *Glot international* 5 (2002).

[20] Dwight Bolinger. "Yes—no questions are not alternative questions". In: *Questions.* Springer, 1978, pp. 87–105.

[21] Antoine Bordes, Y-Lan Boureau, and Jason Weston. "Learning end-to-end goal-oriented dialog". In: *arXiv preprint arXiv:1605.07683* (2016).

[22] Caroline Bousquet-Vernhettes, Régis Privat, and Nadine Vigouroux. "Error handling in spoken dialogue systems: toward corrective dialogue". In: *ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems.* 2003.

[23] Susan E Brennan. "The grounding problem in conversations with and through computers". In: *Social and cognitive approaches to interpersonal communication* (1998), pp. 201–225.

[24] Gerhard Brewka, Hannes Strass, Stefan Ellmauthaler, Johannes Peter Wallner, and Stefan Woltran. "Abstract dialectical frameworks revisited". In: *Twenty-Third International Joint Conference on Artificial Intelligence.* 2013.

[25] Gerhard Brewka and Stefan Woltran. "Abstract dialectical frameworks". In: *Twelfth International Conference on the Principles of Knowledge Representation and Reasoning.* 2010.

[26] Penelope Brown and Stephen C Levinson. "Universals in language usage: Politeness phenomena". In: *Questions and politeness: Strategies in social interaction.* Cambridge University Press, 1978, pp. 56–311.

[27] Harry Bunt, Jan Alexandersson, Jean Carletta, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Kiyong Lee, Volha Petukhova, Andrei Popescu-Belis, Laurent Romary, et al. "Towards an ISO standard for dialogue act annotation". In: *Seventh conference on International Language Resources and Evaluation (LREC'10).* 2010.

[28] Harry Bunt and William Black. *Abduction, Belief and Context in Dialogue: Studies in Computational Pragmatics.* Vol. 1. John Benjamins Publishing, 2000.

[29] Daniel Buring and Christine Gunlogson. *Aren't positive and negative polar questions the same?* 2000.

[30] Hendrik Buschmeier. "Attentive Speaking. From Listener Feedback to Interactive Adaptation". In: (2018).

[31] Hendrik Buschmeier and Stefan Kopp. “Communicative listener feedback in human–agent interaction: artificial speakers need to be attentive and adaptive”. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 2018.

[32] Hendrik Buschmeier and Stefan Kopp. “Communicative Listener Feedback in Human-Agent Interaction: Artificial Speakers Need to Be Attentive and Adaptive”. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS ’18. Stockholm, Sweden: International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 1213–1221. URL: `http://dl.acm.org/citation.cfm?id=3237383.3237880`.

[33] Mary Sandra Carberry. “Pragmatic Modeling in Information System Interfaces (Goals, Dialogue, Plans, Ill-formedness)”. PhD thesis. Newark, DE, USA, 1985.

[34] Peter Carruthers and Peter K Smith. *Theories of theories of mind*. Cambridge University Press, 1996.

[35] Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso. “EVALITA 2018: Overview of the 6th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian”. In: *Proceedings of Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018)*. Ed. by Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso. Turin, Italy: CEUR.org, 2018.

[36] Eve V. Clark. “Common Ground”. In: *The Handbook of Language Emergence*. Chichester, UK: Wiley, 2015, pp. 328–353. DOI: `10.1002/9781118346136.ch15`.

[37] Herbert H. Clark. *Using Language*. Cambridge, UK: Cambridge University Press, 1996. DOI: `10.1017/CBO9780511620539`.

[38] Herbert H. Clark and Susan E. Brennan. “Grounding in communication”. In: *Perspectives on Socially Shared Cognition*. Ed. by Lauren B. Resnick, John M. Levine, and Stephanie D. Teasley. Washington, DC, USA: American Psychological Association, 1991, pp. 222–233.

[39] Herbert H Clark and Edward F Schaefer. “Contributing to discourse”. In: *Cognitive science* 13.2 (1989), pp. 259–294.

[40] Herbert H Clark and Deanna Wilkes-Gibbs. “Referring as a collaborative process”. In: *Cognition* 22.1 (1986), pp. 1–39.

[41] Jacob Cohen. “A Coefficient of Agreement for Nominal Scales”. In: *Educational and Psychological Measurement* 20.1 (1960), pp. 37–46.

[42] Elizabeth Couper-Kuhlen. “Some truths and untruths about final intonation in conversational questions”. In: *Questions: Formal, functional and interactional perspectives*. Cambridge University Press, 2012.

[43] Francesco Cutugno, Felice Dell’Orletta, Isabella Poggi, Renata Savy, and Antonio Sorgente. “The CHROME Manifesto: integrating multimodal data into Cultural Heritage Resources”. In: *Fifth Italian Conference on Computational Linguistics, CLiC-it*. 2018.

[44] Pitt David. “Mental representation”. In: *Retrieved Nov* 15 (2013), p. 2015.

[45] Marco De Boni and Suresh Manandhar. “Implementing clarification dialogues in open domain question answering”. In: *Natural Language Engineering* 11.4 (2005), pp. 343–362.

[46] Alessandro Del Sole. “Introducing microsoft cognitive services”. In: *Microsoft Computer Vision APIs Distilled*. Springer, 2018, pp. 1–4.

[47] Maria Di Maro, Mohamed Diaoulé Diallo, and Francesco Cutugno. “Information-Processing Machines and the Access-Conscious Recognition of Common Ground Inconsistencies: A Proposal.” In: *Proceedings of the Second Symposium on Psychology-based Technologies* ().

[48] Maria Di Maro, Sara Falcone, and Francesco Cutugno. “Prosodic Analysis in Human-Machine Interaction”. In: *Studi AISV* 1 (2018), pp. 227–239.

[49] Maria Di Maro, Antonio Origlia, and Francesco Cutugno. “Cutting melted butter? Common Ground inconsistencies management in dialogue systems using graph databases”. In: *Italian Journal of Computational Linguistics, Special Issue on Computational Dialogue Modelling* (2021), *submitted.*

[50] Maria Di Maro, Antonio Origlia, and Francesco Cutugno. “Overview of the EVALITA 2018 Spoken Utterances Guiding Chef’s Assistant Robots (SUGAR) Task”. In: *Proceedings of EVALITA 2018*. 2018.

[51] Maria Di Maro, Antonio Origlia, and Francesco Cutugno. “PolarExpress: Polar question forms expressing bias-evidence conflicts in Italian”. In: *International Journal of Linguistics* (2021), *submitted.*

[52] Maria Di Maro, Antonio Origlia, and Francesco Cutugno. “Solving Common Ground inconsistencies: the role of polar question forms in human-machine interaction”. In: *Computer Speech and Language* (2021), *submitted.*

[53] Maria Di Maro, Marco Valentino, Anna Riccio, and Antonio Origlia. “Graph Databases for Designing High-Performance Speech Recognition Grammars”. In: *IWCS 2017. 12th International Conference on Computational Semantics—Short papers*. 2017.

[54] Oxford English Dictionary. “Oxford english dictionary”. In: *Simpson, JA & Weiner, ESC* (1989).

[55] Alan Dix, Alan John Dix, Janet Finlay, Gregory D Abowd, and Russell Beale. *Human-computer interaction*. Pearson Education, 2003.

[56] Filippo Domaneschi, Maribel Romero, and Bettina Braun. “Bias in polar questions: Evidence from English and German production experiments”. In: *Glossa: a Journal of General Linguistics* 2.1 (2017).

[57] Matthew S. Dryer. “Polar Questions”. In: *The World Atlas of Language Structures Online*. Ed. by Matthew S. Dryer and Martin Haspelmath. Leipzig: Max Planck Institute for Evolutionary Anthropology, 2013. URL: `https://wals.info/chapter/116`.

[58] Phan Minh Dung. “On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games”. In: *Artificial intelligence* 77.2 (1995), pp. 321–357.

[59] Charles J Fillmore. “Frame semantics”. In: *Cognitive Linguistics: Basic Readings* 34 (2006), pp. 373–400.

[60] Chris D Frith and Uta Frith. “The neural basis of mentalizing”. In: *Neuron* 50.4 (2006), pp. 531–534.

[61] 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.

[62] Olga K Garnica and Martha L King. *Language, children and society: the effect of social factors on children learning to communicate.* Elsevier, 2014.

[63] Jonathan Ginzburg. “Clarifying utterances”. In: *Proc. of the twente workshop on the formal semantics and pragmatics of dialogues. Universiteit Twente, Faculteit Informatica, Enschede.* Citeseer. 1998, pp. 11–30.

[64] Jonathan Ginzburg and Robin Cooper. “Resolving ellipsis in clarification”. In: *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics.* 2001, pp. 236–243.

[65] Jonathan Ginzburg and Zoran Macura. “The emergence of metacommunicative interaction: Some theory, some practice”. In: *Proceedings of the 2nd International Symposium on the Emergence and Evolution of Linguistic Communication.* Hatfield, UK, 2005, pp. 35–40.

[66] Toni Giorgino. “Computing and visualizing dynamic time warping alignments in R: the dtw package”. In: *Journal of Statistical Software* 31.7 (2009), pp. 1–24.

[67] Erving Goffman. “On face-work”. In: *Interaction ritual* (1967), pp. 5–45.

[68] Linda S Gottfredson. *Mainstream science on intelligence: An editorial with 52 signatories, history, and bibliography.* Citeseer, 1997.

[69] Herbert Paul Grice. “Logic and conversation”. In: *Speech acts.* Brill, 1975, pp. 41–58.

[70] Michael Alexander Kirkwood Halliday. “Notes on transitivity and theme in English: Part 3”. In: *Journal of linguistics* 4.2 (1968), pp. 179–215.

[71] A Handbook. “From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity”. In: (2003).

[72] Kaoru Hayano. “19 Question Design in Conversation”. In: *The handbook of conversation analysis* (2013), p. 395.

[73] John Heritage. “The limits of questioning: Negative interrogatives and hostile question content”. In: *Journal of Pragmatics* 34.10-11 (2002), pp. 1427–1446.

[74] Graeme Hirst, Susan McRoy, Peter Heeman, Philip Edmonds, and Diane Horton. “Repairing conversational misunderstandings and non-understandings”. In: *Speech Communication* 15.3-4 (1994), pp. 213–229.

[75] Julian Hough, Sina Zarrieß, and David Schlangen. “Grounding Imperatives to Actions is Not Enough: A Challenge for Grounded NLU for Robots from Human-Human data”. In: *GLU 2017 International Workshop on Grounding Language Understanding.* Aug. 2017, pp. 88–91. DOI: 10.21437/GLU.2017-18.

[76] Yan Huang. *The Oxford Handbook of Pragmatics.* Oxford University Press, 2017.

[77] Andrew Hunt and Scott McGlashan. “Speech recognition grammar specification version 1.0”. In: *W3C Recommendation, March* (2004).

[78] Dell Hymes. “On communicative competence”. In: *Sociolinguistics* 269293 (1972), pp. 269–293.

[79] Roman Jakobson. “Metalanguage as a linguistic problem”. In: *Selected writings* 7 (1956), pp. 113–121.

[80] Veton Këpuska and Gamal Bohouta. “Comparing speech recognition systems (Microsoft API, Google API and CMU Sphinx)”. In: *International Journal of Engineering Research and Application* 7.03 (2017), pp. 20–24.

[81] Sangeet S Khemlani and Philip N Johnson-Laird. “Hidden conflicts: Explanations make inconsistencies harder to detect”. In: *Acta Psychologica* 139.3 (2012), pp. 486–491.

[82] Thomas Kisler, Uwe Reichel, and Florian Schiel. “Multilingual processing of speech via web services”. In: *Computer Speech & Language* 45 (2017), pp. 326–347.

[83] Thomas Kisler, Florian Schiel, and Han Sloetjes. “Signal processing via web services: the use case WebMAUS”. In: *Digital Humanities Conference 2012*. 2012.

[84] Eric M Kok, John-Jules Ch Meyer, Henry Prakken, and Gerard AW Vreeswijk. “A formal argumentation framework for deliberation dialogues”. In: *International Workshop on Argumentation in Multi-Agent Systems*. Springer. 2010, pp. 31–48.

[85] Irene Koshik. “A conversation analytic study of yes/no questions which convey reversed polarity assertions”. In: *Journal of Pragmatics* 34.12 (2002), pp. 1851–1877.

[86] Irene Koshik. *Beyond rhetorical questions: Assertive questions in everyday interaction*. Vol. 16. John Benjamins Publishing, 2005.

[87] Spyros Kousidis, Casey Kennington, Timo Baumann, Hendrik Buschmeier, Stefan Kopp, and David Schlangen. “A Multimodal In-Car Dialogue System That Tracks The Driver’s Attention”. In: *Proceedings of the 16th International Conference on Multimodal Interaction*. ACM. 2014, pp. 26–33.

[88] Manfred Krifka. “Negated polarity questions as denegations of assertions”. In: *Contrastiveness in information structure, alternatives and scalar implicatures*. Springer, 2017, pp. 359–398.

[89] William H Kruskal and W Allen Wallis. “Use of ranks in one-criterion variance analysis”. In: *Journal of the American statistical Association* 47.260 (1952), pp. 583–621.

[90] William Labov and David Fanshel. *Therapeutic discourse: Psychotherapy as conversation*. Academic Press, 1977.

[91] Dwight Robert Ladd. “A first look at the semantics and pragmatics of negative questions and tag questions”. In: *Papers from the Regional Meeting. Chicago Ling. Soc. Chicago, Ill.* 17. 1981, pp. 164–171.

[92] Geoffrey Leech. “Pragmatics and Dialogue”. In: *The Oxford Handbook of Computational Linguistics*. 2003.

[93] Alessandro Lenci, Simonetta Montemagni, and Vito Pirrelli. *Testo e computer. Introduzione alla linguistica computazionale*. Carocci editore, 2005.

[94] Stephen C Levinson. “Interactional biases in human thinking”. In: *Social intelligence and interaction*. Cambridge University Press, 1995, pp. 221–260.

[95] Stephen C Levinson. “Questions and responses in Yélı Dnye, the Papuan language of Rossel Island”. In: *Journal of Pragmatics* 42.10 (2010), pp. 2741–2755.

[96] Marcin Lewiński. “Argumentative Discussion: The Rationality of What?” In: *Topoi* 38 (Dec. 2019). DOI: 10.1007/s11245-015-9361-0.

[97] Marcin Lewiński. “Argumentative discussion: The rationality of what?” In: *Topoi* 38.4 (2019), pp. 645–658.

[98] Pierre Lison and Casey Kennington. “Opendial: A toolkit for developing spoken dialogue systems with probabilistic rules”. In: *Proceedings of ACL-2016 system demonstrations*. 2016, pp. 67–72.

[99] Gustavo López, Luis Quesada, and Luis A Guerrero. “Alexa vs. Siri vs. Cortana vs. Google Assistant: a comparison of speech-based natural user interfaces”. In: *International Conference on Applied Human Factors and Ergonomics*. Springer. 2017, pp. 241–250.

[100] Andy Lücking, Kirsten Bergman, Florian Hahn, Stefan Kopp, and Hannes Rieser. “Data-based analysis of speech and gesture: the Bielefeld Speech and Gesture Alignment corpus (SaGA) and its applications”. In: *Journal on Multimodal User Interfaces* 7 (Aug. 2012), pp. 5–18. DOI: 10.1007/s12193-012-0106-8.

[101] Andy Lücking, Kirsten Bergmann, Florian Hahn, Stefan Kopp, and Hannes Rieser. “The Bielefeld Speech and Gesture Alignment corpus (SaGA)”. In: *LREC 2010 workshop: Multimodal corpora–advances in capturing, coding and analyzing multimodality*. 2010.

[102] William G Lycan. *Consciousness and experience*. Mit Press, 1996.

[103] Fabrizio Macagno and Sarah Bigi. “Analyzing the Pragmatic Structure of Dialogues”. In: *Discourse Studies* 19.2 (2017), pp. 148–168.

[104] Markéta Malá. “Negative polar questions in English and Czech”. In: *Proceedings of the 4lh Corpus Linguis tics Conference*. 2007.

[105] Giorgio Marchetti. “Consciousness: a unique way of processing information”. In: *Cognitive processing* 19.3 (2018), pp. 435–464.

[106] Scott McGlashan, Norman Fraser, Nigel Gilbert, Eric Bilange, Paul Heisterkamp, and Nick Youd. “Dialogue management for telephone information systems”. In: *Proceedings of the third conference on Applied natural language processing*. Association for Computational Linguistics. 1992, pp. 245–246.

[107] Helle Metslang, Külli Habicht, and Karl Pajusalu. “Where do polar question markers come from?” In: *STUF-Language Typology and Universals* 70.3 (2017), pp. 489–521.

[108] Meinard Müller. “Dynamic time warping”. In: *Information retrieval for music and motion* (2007), pp. 69–84.

[109] Romy Müller, Dennis Paul, and Yijun Li. “Reformulation of symptom descriptions in dialogue systems for fault diagnosis: How to ask for clarification?” In: *International Journal of Human-Computer Studies* 145 (), p. 102516.

[110] Antonio Origlia, Francesco Cutugno, Antonio Rodà, Piero Cosi, and Claudio Zmarich. “FANTASIA: a framework for advanced natural tools and applications in social, interactive approaches”. In: *Multimedia Tools and Applications* 78.10 (2019), pp. 13613–13648.

[111] Antonio Origlia, Renata Savy, Isabella Poggi, Francesco Cutugno, Iolanda Alfano, Francesca D'Errico, Laura Vincze, and Violetta Cataldo. "An Audiovisual Corpus of Guided Tours in Cultural Sites: Data Collection Protocols in the CHROME Project". In: *Proceedings of the 2018 AVI-CH Workshop on Advanced Visual Interfaces for Cultural Heritage*. Vol. 2091. 2018.

[112] Riccardo Orrico. "Individual variability in intonational meaning identification: The role of cognitive and sociolinguistic variables". PhD thesis. Salerno, Italy: Università degli studi di Salerno, Aix Marseille Université, 2020.

[113] Riccardo Orrico, Renata Savy, and Mariapaola D'Imperio. "Salerno Italian: Intonational phonology and dimensions of variation". In: *Gli archivi sonori al crocevia tra scienze fonetiche, informatica umanistica e patrimonio digitale [Audio archives at the crossroads of speech sciences, digital humanities and digital heritage] Studi AISV* 6 (2019).

[114] David Yoshikazu Oshima. "Remarks on epistemically biased questions". In: *Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation*. 2017, pp. 169–177.

[115] Miriam RL Petruck. "Frame semantics". In: *Handbook of pragmatics* 1 (1996), p. 13.

[116] Piotr Pezik. "Spokes-a search and exploration service for conversational corpus data". In: *Selected Papers from the CLARIN 2014 Conference, October 24-25, 2014, Soesterberg, The Netherlands*. Linköping University Electronic Press. 2015, pp. 99–109.

[117] John L Pollock. "Defeasible reasoning". In: *Cognitive science* 11.4 (1987), pp. 481–518.

[118] Henry Prakken. "A formal model of adjudication dialogues". In: *Artificial Intelligence and Law* 16.3 (2008), pp. 305–328.

[119] Henry Prakken. "Coherence and flexibility in dialogue games for argumentation". In: *Journal of logic and computation* 15.6 (2005), pp. 1009–1040.

[120] Henry Prakken. *Historical overview of formal argumentation*. Vol. 1. College Publications, 2018.

[121] Matthew Purver. "Clarie: The clarification engine". In: *Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue (Catalog)*. Citeseer. 2004, pp. 77–84.

[122] Matthew Purver. "The Theory and Use of Clarification Requests in Dialogue". PhD thesis. London, UK: King's College, University of London, 2004.

[123] Matthew Purver, Jonathan Ginzburg, and Patrick Healey. "On the means for clarification in dialogue". In: *Current and new directions in discourse and dialogue*. Springer, 2003, pp. 235–255.

[124] Verena Rieser, Ivana Kruijff-Korbayová, and Oliver Lemon. "A corpus collection and annotation framework for learning multimodal clarification strategies". In: *6th SIGdial Workshop on DISCOURSE and DIALOGUE*. 2005.

[125] Verena Rieser and Johanna D Moore. "Implications for generating clarification requests in task-oriented dialogues". In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. Ann Arbor, Michigan, 2005, pp. 239–246. DOI: 10.3115/1219840.1219870.

[126] Alan Ritter, Colin Cherry, and Bill Dolan. “Unsupervised modeling of twitter conversations”. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. 2010, pp. 172–180.

[127] Kepa Joseba Rodrıguez and David Schlangen. “Form, intonation and function of clarification requests in German task-oriented spoken dialogues”. In: *Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue*. Barcelona, Catalonia, Spain, 2004.

[128] Floris Roelofsen, Noortje Venhuizen, and Galit Weidman Sassoon. “Positive and negative questions in discourse”. In: *Proceedings of Sinn und Bedeutung*. Vol. 17. 2013, pp. 455–472.

[129] Maribel Romero and Chung-hye Han. “On negative yes/no questions”. In: *Linguistics and philosophy* 27.5 (2004), pp. 609–658.

[130] Antonio Roque and David Traum. “Improving a virtual human using a model of degrees of grounding”. In: *Twenty-First International Joint Conference on Artificial Intelligence*. Citeseer. 2009.

[131] Adriana Roventini, Antonietta Alonge, Nicoletta Calzolari, Bernardo Magnini, and Francesca Bertagna. “ItalWordNet: a Large Semantic Database for Italian.” In: *LREC*. 2000.

[132] T Ruffman, L Slade, and M Taumoepeau. “Theory of mind and language ability: Understanding the bigger picture”. In: *Invited paper, web conference on coevolution of language and theory of mind. Retrieved from*. Vol. 6. 21. 2004, p. 2007.

[133] Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. “A simplest systematics for the organization of turn taking for conversation”. In: *Studies in the organization of conversational interaction*. Elsevier, 1978, pp. 7–55.

[134] Andrew Sanders. *An introduction to Unreal engine 4*. CRC Press, 2016.

[135] Michelina Savino and Martine Grice. “The perception of negative bias in Bari Italian questions”. In: *Prosodic categories: Production, perception and comprehension*. Springer, 2011, pp. 187–206.

[136] Renata Savy. “Pr. A. Ti. D: A Coding Scheme for Pragmatic Annotation of Dialogues.” In: *LREC*. 2010.

[137] Renata Savy and Francesco Cutugno. “CLIPS: diatopic, diamesic and diaphasic variations of spoken Italian”. In: *Proceedings of the Corpus Linguistics Conference 2009 (CL2009),* 2009, p. 213.

[138] Jost Schatzmann, Kallirroi Georgila, and Steve Young. “Quantitative evaluation of user simulation techniques for spoken dialogue systems”. In: *6th SIGdial Workshop on DISCOURSE and DIALOGUE*. 2005.

[139] Florian Schiel. “Automatic phonetic transcription of non-prompted speech”. In: *Proc. of the ICPhS* (1999), pp. 607–610.

[140] David Schlangen. “Causes and strategies for requesting clarification in dialogue”. In: *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue at HLT-NAACL 2004*. 2004, pp. 136–143.

[141] Julian J Schlöder and Raquel Fernández. “Clarifying intentions in dialogue: A corpus study”. In: *Proceedings of the 11th International Conference on Computational Semantics*. 2015, pp. 46–51.

[142] Helmut Schmid, Marco Baroni, Eros Zanchetta, and Achim Stein. “The enriched treetagger system”. In: *proceedings of the EVALITA 2007 workshop*. 2007.

[143] Thomas Schmidt and Kai Wörner. “EXMARaLDA–Creating, analysing and sharing spoken language corpora for pragmatic research”. In: *Pragmatics. Quarterly Publication of the International Pragmatics Association (IPrA)* 19.4 (2009), pp. 565–582.

[144] Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. “A Survey of Available Corpora For Building Data-Driven Dialogue Systems: The Journal Version”. In: *Dialogue & Discourse* 9.1 (2018), pp. 1–49.

[145] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. “Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models.” In: *AAAI*. Vol. 16. 2016, pp. 3776–3784.

[146] Claude E. Shannon and Warren Weaver. *The Mathematical Theory of Communication*. 1949.

[147] Samuel Sanford Shapiro and Martin B Wilk. “An analysis of variance test for normality (complete samples)”. In: *Biometrika* 52.3/4 (1965), pp. 591–611.

[148] Jack Sidnell. ““Who knows best?”: Evidentiality and epistemic asymmetry in conversation”. In: *Pragmatics and Society* 3.2 (2012), pp. 294–320.

[149] Jack Sidnell and Tanya Stivers. *The Handbook of Conversation Analysis*. Vol. 121. John Wiley & Sons, 2012.

[150] Gabriel Skantze. “Exploring human error recovery strategies: Implications for spoken dialogue systems”. In: *Speech Communication* 45.3 (2005), pp. 325–341.

[151] Aaron Sloman and Ron Chrisley. “Virtual machines and consciousness”. In: *Journal of Consciousness Studies* 10.4-5 (2003), pp. 133–172.

[152] Thorvald Sorensen. “A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons”. In: *Biologiske Skrifter* 5 (1948), pp. 1–34.

[153] Dan Sperber, Fabrice Clément, Christophe Heintz, Olivier Mascaro, Hugo Mercier, Gloria Origgi, and Deirdre Wilson. “Epistemic vigilance”. In: *Mind & Language* 25.4 (2010), pp. 359–393.

[154] Dan Sperber and Deirdre Wilson. *Experimental Pragmatics*. Citeseer, 2004.

[155] Robert Stalnaker. “Common ground”. In: *Linguistics and philosophy* 25.5/6 (2002), pp. 701–721.

[156] Robert Stalnaker. “Pragmatic presuppositions”. In: *Proceedings of the Texas conference on per˜ formatives, presuppositions, and implicatures. Arlington, VA: Center for Applied Linguistics*. ERIC. 1977, pp. 135–148.

[157] Sonja Stange and Stefan Kopp. “Effects of a Social Robot’s Self-Explanations on How Humans Understand and Evaluate Its Behavior”. In: *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. 2020, pp. 619–627.

[158] Jakob Steensig and Paul Drew. *Questioning and Affiliation/Disaffiliation in Interaction: Special Issue of Discourse Processes*. SAGE Publications, 2008.

[159] Tanya Stivers and Nick J Enfield. "A coding scheme for question–response sequences in conversation". In: *Journal of Pragmatics* 42.10 (2010), pp. 2620–2626.

[160] Sarah Stowell. *Using R for Statistics*. Apress, 2014. ISBN: 978-1484201404. URL: `http://www.apress.com/9781484201404`.

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

[162] NIST Speech Recognition Scoring Toolkit. "Speech recognition scoring toolkit". In: (2001).

[163] David R. Traum. "Computational models of grounding in collaborative systems". In: *Psychological Models of Communication in Collaborative Systems: Papers from the AAAI Fall Symposium*. North Falmouth, MA, USA, 1999, pp. 124–131.

[164] Gokhan Tur and Renato De Mori. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

[165] Gokhan Tur and Li Deng. "Intent determination and spoken utterance classification". In: *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech* (2011), pp. 93–118.

[166] Alan Mathison Turing. *Computing Machinery and Intelligence*.

[167] Oriol Vinyals and Quoc Le. "A neural conversational model". In: *arXiv preprint arXiv:1506.05869* (2015).

[168] Miriam Voghera and Francesco Cutugno. "AN. ANA. S.: aligning text to temporal syntagmatic progression in Treebanks". In: *Proceedings of the 5th Corpus Linguistics Conference, Liverpool*. 2009, pp. 20–23.

[169] Klaus Von Heusinger and Christoph Schwarze. "Underspecification in the semantics of word formation: the case of denominal verbs of removal in Italian". In: *Linguistics* 44.6 (2006), pp. 1165–1194.

[170] Michaela M Wagner-Menghin. "Binomial Test". In: *Wiley StatsRef: Statistics Reference Online* (2014).

[171] Douglas N Walton. *Logical Dialogue-Games*. University Press of America, Lanham, Maryland, 1984.

[172] Douglas Walton and David M Godden. "The impact of argumentation on artificial intelligence". In: *Considering Pragma-Dialectics, Mahwah, Erlbaum, New Jersey* (2006), pp. 287–299.

[173] Douglas Walton and Erik CW Krabbe. *Commitment in dialogue: Basic concepts of interpersonal reasoning*. SUNY press, 1995.

[174] Paul Watzlawick, Janet Helmick Beavin, and Don D Jackson. "Pragmatics of human communication: A study of interactional patterns". In: *Pathologies, and Paradoxes Chapter: Psychotherapy* (1967).

[175] Jim Webber and Ian Robinson. *A programmatic introduction to neo4j*. Addison-Wesley Professional, 2018.

[176] Joseph Weizenbaum. "ELIZA—a computer program for the study of natural language communication between man and machine". In: *Communications of the ACM* 9.1 (1966), pp. 36–45.

[177] Frank Wilcoxon. “Individual comparisons by ranking methods”. In: *Breakthroughs in Statistics*. Springer, 1992, pp. 196–202.

[178] Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann, and Han Sloetjes. “ELAN: a professional framework for multimodality research”. In: *5th International Conference on Language Resources and Evaluation (LREC 2006)*. 2006, pp. 1556–1559.

[179] Tangming Yuan, David Moore, and Alec Grierson. “Human-computer debate, a computational dialectics approach”. In: *unpublished doctoral dissertation, Leeds Metropolitan University* (2004).