

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

RODRIGO SOUZA WILKENS

**A study of the use of Natural Language
Processing for Conversational Agents**

Dissertação apresentada como requisito parcial para
a obtenção do grau de Mestre em Ciência da
Computação

Orientador: Profa. Dra. Aline Villavicencio
Co-orientador: Profa. Dra. Rosa Vicari

Porto Alegre
2016

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Wilkens, Rodrigo Souza

A study of the use of Natural Language Processing for Conversational Agents / Rodrigo Souza Wilkens. – Porto Alegre: da UFRGS, 2016.

83 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul– Programa de Pós-Graduação em Computação. , Porto Alegre, BR–RS, 2016. Orientador: Aline Villavicencio; Co-orientador: Rosa Vicari.

1. Natural Language Processing. 2. Conversational Agents. 3. Natural Language Understanding. 4. Parser. 5. Combinatory Categorical Grammar. 6. Grammar acquisition. 7. Portuguese. I. Villavicencio, Aline. II. Vicari, Rosa. III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Pós-Graduação: Prof. Vladimir Pinheiro do Nascimento

Diretor do Instituto de Informática: Prof. Luis da Cunha Lamb

:

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

ACKNOWLEDGEMENTS

I'd like to thank my family, specially my fiancée Anike, without whose love, support and friendship, this work would not have been finished. Thanks also to Aline and Rosa to advise which was crucial for elaboration of this work, through discussions about this and in indicating ways; to Comunica project members, especially to Leonardo who made a great contribution to quality of this work.

SUMÁRIO

LISTA DE FIGURAS	6
LISTA DE TABELAS	7
LISTA DE ABREVIATURAS E SIGLAS	8
RESUMO	9
ABSTRACT	10
1 INTRODUCTION.....	12
2 CONVERSATIONAL AGENTS.....	16
2.1 Natural Language Understanding.....	16
2.1.1 A linguistic perspective.....	16
2.1.2 Approaches to Computational Understanding.....	18
2.2 Dialogue	21
2.2.1 The Dialogue Features	21
2.2.2 Chatterbots	26
2.2.3 Conversational Systems	27
2.2.4 Beliefs, Desires and Intentions (BDI).....	29
2.3 Summary.....	31
3 QUESTION ANSWERING	32
3.1 The Internal Steps.....	33
3.2 Question Answering Systems.....	36
3.3 Question Answering for Portuguese.....	37
3.4 Summary.....	39
4 RESOURCES AND TOOLS.....	41
4.1 Morphosyntactic Information.....	41
4.2 Combinatory Categorical Grammar (CCG).....	42
4.3 Corpus resources.....	44
4.4 Ontology.....	45
5 BASE SYSTEMS.....	47
5.1 Question Answering for Portuguese: how much is needed?.....	47
5.1.1 Materials and Methods.....	47
5.1.2 Results.....	49
5.2 COMUNICA - A Question Answering System for Brazilian Portuguese.....	51
5.2.1 Speech Recognition	51
5.2.2 Text Processing	52
5.2.3 Database Access.....	52
5.2.4 Speech Synthesis.....	53
5.2.5 Manager	53
5.3 Grammar Engineering	53
5.3.1 Algorithm.....	54
5.3.2 Parser Evaluation	56
5.3.2.1 Statistical Evaluation	56
5.3.2.2 Human Evaluation	59
5.4 Summary.....	59
6 ARCHITECTURE.....	63
6.1 Conversational Plans	63
6.2 The Proposed Alice	64
6.3 The Proposed Question Answering System.....	65
6.4 Task Manager	66
6.5 Summary.....	66

7 EVALUATION.....	68
7.1 Dialogue	68
7.2 Summary.....	71
8 CONCLUSIONS AND FUTURE WORK	72
8.1 Conclusions.....	72
8.2 Future Work	73
APÊNDICE.....	75
REFERÊNCIAS.....	79

LISTA DE FIGURAS

Figura 1.1	General architecture of a conversational agent.....	14
Figura 2.1	The virtual human system architecture.....	22
Figura 2.2	Illustration of the three techniques to managing computational dialogue.....	25
Figura 2.3	The semantic representation of the utterance “the building is made of wood”.	28
Figura 3.1	Example of QA stages	33
Figura 3.2	Example of extraction of information approaches.....	35
Figura 3.3	Performance of QA systems for Portuguese, QA@CLEF.....	40
Figura 4.1	CCG derivation history	43
Figura 4.2	An example of Bosque corpus (original).....	45
Figura 4.3	An example of the Bosque corpus in a classic tree format.....	45
Figura 4.4	Ontologie of FAMURS domain.....	46
Figura 5.1	System architecture developed.	48
Figura 5.2	Architecture of the system.	51
Figura 5.3	The domain ontology	60
Figura 5.4	An example of Bosque corpus.....	61
Figura 5.5	Example of the 4 steps of corpus reformatting	61
Figura 5.6	Example of direct allocation step	61
Figura 5.7	Example of indirect allocation.....	61
Figura 5.8	Comparison of learning category combination rules from 9 fragments	62
Figura 5.9	Learning curve of PoS	62
Figura 5.10	Frequency of class after evaluation manual.....	62
Figura 6.1	Example of dialogue and system internal messages.....	64

LISTA DE TABELAS

Tabela 2.1 Searly taxonomy	24
Tabela 3.1 Rules for identifying the type of response expected.....	34
Tabela 5.1 Percentage of correct answers per type of questions	49
Tabela 5.2 Performance of the document retrieval module.....	50
Tabela 5.3 Results of factoid and non-factoids question and their respective methods	50
Tabela 5.4 Statistical evaluation of the parser	57
Tabela 5.5 Numbers of PoS resulting of manual evaluation of the categories quality.....	58
Tabela 6.1 Five Patterns	64
Tabela 7.1 Major categories of Embodied Conversational Agent research	69
Tabela 7.2 Explicit Dialogue Evaluation.....	70

LISTA DE ABREVIATURAS E SIGLAS

PoS	part-of-speech
NLU	Natural Language Understanding
CCG	Combinatory Categorical Grammar
BDI	Belief, Desire and Intention
NL	Natural Language
AIML	Artificial Intelligence Markup Language

RESUMO

Language is a mark of humanity and conscience, with the conversation (or dialogue) as one of the most fundamental manners of communication that we learn as children. Therefore one way to make a computer more attractive for interaction with users is through the use of natural language. Among the systems with some degree of language capabilities developed, the Eliza chatterbot is probably the first with a focus on dialogue.

In order to make the interaction more interesting and useful to the user there are other approaches besides chatterbots, like conversational agents. These agents generally have, to some degree, properties like: a body (with cognitive states, including beliefs, desires and intentions or objectives); an interactive incorporation in the real or virtual world (including perception of events, communication, ability to manipulate the world and communicate with others); and behavior similar to a human (including affective abilities). This type of agents has been called by several terms, including animated agents or embedded conversational agents (ECA).

A dialogue system has six basic components. (1) The speech recognition component is responsible for translating the user's speech into text. (2) The Natural Language Understanding component produces a semantic representation suitable for dialogues, usually using grammars and ontologies. (3) The Task Manager chooses the concepts to be expressed to the user. (4) The Natural Language Generation component defines how to express these concepts in words. (5) The dialog manager controls the structure of the dialogue. (6) The synthesizer is responsible for translating the agents answer into speech.

However, there is no consensus about the necessary resources for developing conversational agents and the difficulties involved (especially in resource-poor languages). This work focuses on the influence of natural language components (dialogue understander and manager) and analyses, in particular the use of parsing systems as part of developing conversational agents with more flexible language capabilities. This work analyses what kind of parsing resources contributes to conversational agents and discusses how to develop them targeting Portuguese, which is a resource-poor language. To do so we analyze approaches to the understanding of natural language, and identify parsing approaches that offer good performance, based on which we develop a prototype to evaluate the impact of using a parser in a conversational agent.

Palavras-chave: Natural Language Processing. Conversational Agents. Natural Language Understanding. Parser. Combinatory Categorical Grammar. Grammar acquisition. Portuguese.

Um estudo do uso de Processamento de Linguagem Natural em Agentes Conversacionais

ABSTRACT

A linguagem é uma marca da humanidade e da consciência, sendo a conversação (ou diálogo) uma das maneiras de comunicação mais fundamentais que aprendemos quando crianças. Por isso uma forma de fazer um computador mais atrativo para interação com usuários é usando linguagem natural. Dos sistemas com algum grau de capacidade de linguagem desenvolvidos, o chatterbot Eliza é, provavelmente, o primeiro sistema com foco em diálogo.

Com o objetivo de tornar a interação mais interessante e útil para o usuário há outras aplicações além de chatterbots, como agentes conversacionais. Estes agentes geralmente possuem, em algum grau, propriedades como: corpo (com estados cognitivos, incluindo crenças, desejos e intenções ou objetivos); incorporação interativa no mundo real ou virtual (incluindo percepções de eventos, comunicação, habilidade de manipular o mundo e comunicar com outros agentes); e comportamento similar ao humano (incluindo habilidades afetivas). Este tipo de agente tem sido chamado de diversos nomes como agentes animados ou agentes conversacionais incorporados.

Um sistema de diálogo possui seis componentes básicos. (1) O componente de reconhecimento de fala que é responsável por traduzir a fala do usuário em texto. (2) O componente de entendimento de linguagem natural que produz uma representação semântica adequada para diálogos, normalmente utilizando gramáticas e ontologias. (3) O gerenciador de tarefa que escolhe os conceitos a serem expressos ao usuário. (4) O componente de geração de linguagem natural que define como expressar estes conceitos em palavras. (5) O gerenciador de diálogo controla a estrutura do diálogo. (6) O sintetizador de voz é responsável por traduzir a resposta do agente em fala.

No entanto, não há consenso sobre os recursos necessários para desenvolver agentes conversacionais e a dificuldade envolvida nisso (especialmente em línguas com poucos recursos disponíveis). Este trabalho foca na influência dos componentes de linguagem natural (entendimento e gerência de diálogo) e analisa em especial o uso de sistemas de análise sintática (parser) como parte do desenvolvimento de agentes conversacionais com habilidades de linguagem mais flexível. Este trabalho analisa quais os recursos do analisador sintático contribuem para agentes conversacionais e aborda como os desenvolver, tendo como língua alvo o português (uma língua com poucos recursos disponíveis). Para isto, analisamos as abordagens de entendimento de linguagem natural e identificamos as abordagens de análise sintática que oferecem um bom desempenho. Baseados nesta análise, desenvolvemos um protótipo para avaliar o impacto do

uso de analisador sintático em um agente conversacional.

Keywords: Processamento de Linguagem Natural, Agentes Conversacionais, Entendimento de Linguagem Natural, Parser, Combinatory Categorical Grammar, Aquisição de Gramática, Português.

1 INTRODUCTION

Language is a mark of humanity and conscience, with the conversation (or dialogue) as one of the most fundamental manners of communication that we learn as children (JURAFSJY; MARTIN, 2009). A more attractive form of interaction in computational systems with users is through use of natural language. Among the systems developed Eliza (WEIZENBAUM, 1966) is one of the first with a focus on dialogue (WALLIS et al., 2001). Eliza mimics a Rogerian psychologist and has a very simple algorithm, where the text is examined for the presence of keywords and then transformed according to rules (or regular expressions) associated with each keyword. For example:

USER: Men are all alike.

ELIZA: IN WHAT WAY?

USER: They're always bugging us about something or other.

ELIZA: CAN YOU THINK OF A SPECIFIC EXAMPLE?

Retrieved from rules like

IF input = ``I am X''

THEN output = ``How long have you been X?''

With Interpretation

INPUT: ``I am sick''

MATCHES: ``I am X'' where X = ``sick''

RESPONSE: How long have you been sick?

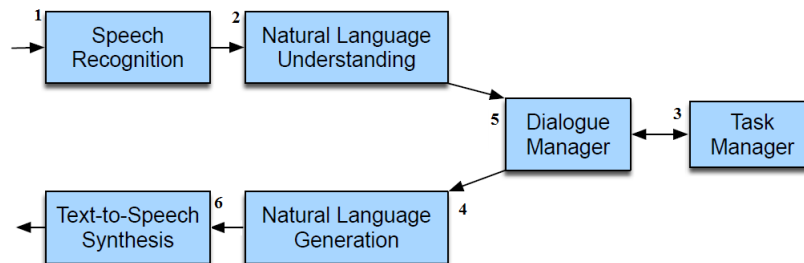
Currently, several processes performed daily by humans are automated in some kind of computer system. In order to overcome the discomfort that some people feel using these types of systems and to allow the interfaces to be increasingly useful and easy to use, embodied interfaces have been developed and customized, represented by human figures or characters (LESTER et al., 2000), (GRATCH; MARSELLA, 2001), (BUISINE; ABRILIAN; MARTIN, 2004) e (GULZ; HAAKE, 2006). These characters and human figures need to try to be what Bates (BATES, 1992) calls lifelike and thus must look smart. In this sense dialogue systems provide a natural language interaction with the user. An example of dialogue system is Galatea (SKANTZE, 2005), which uses tools such as syntactic parsing, speech synthesis and recognition, identification of errors in context, reasoning about uncertainty and semantic knowledge to detect the user's geographic location and point the way to some requested place.

The dialogue systems, however, require tools from areas such as agent architecture, artificial intelligence, speech synthesis, natural language processing, affective computing, computer graphics, human computer interaction, psychology and sociology. Because of this need, most of the systems that have been developed are for English, while for other languages with significantly less resources available there are few basic systems. The problem becomes even more apparent when working with specific domains (e.g. biomedical), employing its own terminology and vocabulary rarely described in dictionaries or ontologies. Moreover, even in languages where such resources are available, there is no consensus on the most appropriate architecture, and what resources are actually needed for a satisfactory communication.

However, more sophisticated mechanisms are needed for a more natural interaction. In this sense there are virtual humans whose goal is an interaction similar to human interaction (CASSEL et al., 2000). To Rockel and Johson (RICKEL; JOHSON, 1999) virtual humans are autonomous agents who can play the role of people in simulations or games. These agents generally have the following properties (at different levels): a body (with cognitive states, including beliefs, desires and intentions or objectives); an interactive incorporation in the real or virtual world (including perception of events, communication, ability to manipulate the world and communicate with others); and behavior similar to a human (including affective abilities). This type of agents has been called by several terms, including animated agents (RICKEL; JOHSON, 1999) or embedded conversational agents (ECA) (CASSEL et al., 2000). With respect to the capacity for dialogue, virtual humans have a number of similarities with both task-oriented dialogue systems and chatterbots. Similarly to task-oriented dialogue systems, these usually have knowledge of the task; and the models of the steps involved in it, and how to talk about this. Usually they focus on resolving a problem as efficiently as possible, even if the dialogues produced sound unnatural to humans, in contrast to virtual humans, where there is an effort to produce a human-like dialogue. Similar to the chatterbot, virtual humans have focused on more believable conversation, but its purpose is not to convince they are human, only to serve as a component to allow people to have a helpful interactive experience (TRAUM et al., 2008).

According to Jurafsky and Martin (JURAFSJKY; MARTIN, 2009), in general, there is no consensus on research goals, methodologies and evaluation to model dialogues, nor about the tools necessary to build a conversational agent. A dialogue system in general has six basic components, such as shown in Figure 1.1. (1) The speech recognition component is responsible for translating the user's speech into text. (2) The component of Natural Language Understanding produces a semantic representation suitable for the task of dialogue, usually using grammars and ontologies. (3) The Task Manager chooses the concepts to be expressed to the user. (4) The

Natural Language Generation component chooses the concepts to be expressed to the user and defines how to express these concepts in words. (5) The dialogue manager component controls the structure of the dialogue. (6) The synthesizer is responsible for translating the agents answer into speech.



Fonte: (JURAFSJY; MARTIN, 2009)

Figura 1.1 – General architecture of a conversational agent

In particular, regarding the Natural Language Understanding component and Dialogue Manager component in this work focus in affects of the natural languages techniques. Considering the lack of consensus on the resources necessary to develop conversational agents and the difficulty of developing them (especially in resource-poor languages), this work focuses on the contribution of parser systems as part of developing conversational agents. This work analyzes what kind of parsing resources contribute to a more flexible and precise interaction with conversational agents and discusses how to develop them.

This work presents the following structure:

- **State of the Art:** Chapters 2 and 3 provide background material for this research looking at important concepts in natural language understanding (NLU), in particular dialogue managers and question answering. We discuss in particular the issues involved in the translation of natural language into a computer interpretable language. In the dialogue manager part, interpretable information (generated by NLU) is related to the agent's actions and this relationship is created through the beliefs that the agent has about the user. Among the actions carried out by the agent, we emphasize using unstructured information to answer questions using a question answering (QA) system.
- **Architecture:** Chapters 4 and 5 describe the materials, methods and architecture employed in this work to support the analysis needed for the conversational agent. We show the translation from natural language to its logical form representation using a Combinatory Categorical Grammar (CCG) based parser. As there is no wide coverage CCG grammar for Portuguese, it is necessary to semi-automatically create one (by machine learning from a corpus). The dialogue manager address the conversational plans, and re-

lates the plans, beliefs and intentions of the agent with actions to perform. The actions which can be performed by the agent include accessing a database, a corpus and a set of conversational responses (e.g. responding to a greeting).

- **Evaluation and Conclusions:** Chapter 6 presents the evaluation performed in order to analyze the influence of the parser in the natural language understanding module of a conversational agent. The analysis has two key points: parsing and dialogue. In the parser evaluation we investigate the coverage capabilities of syntactic parsing over a test corpus (using both human and statistical evaluation). In dialogue evaluation we focus on the quality of dialogue (believability, appropriate handling of the domain and sociability) with an in-depth error-analysis and an investigation of what the performance would be without the parser. In chapter 7 we discuss the conclusions of this work and future directions.

2 CONVERSATIONAL AGENTS

In order to analyze the impact of parsers in conversational agents this chapter discusses the state of the art of the modules required for language understanding and dialogue management. In the understanding of natural language we discuss how systems perform the understanding, as well as the tools usually used for the task of understanding. In dialogue manager we look at the elements of the conversation, possible approaches, and their implementations.

2.1 Natural Language Understanding

The task of Natural Language Understanding (NLU) can be considered as the process of translating natural language into a language interpretable by a computer. This section explains the state of the art in NLU systems that perform this translation (including the representation of predicate logic).

2.1.1 A linguistic perspective

Spoken and written language functions have been analysed by different research areas such as anthropology and sociology, and the written language has two main functions: storing (enabling communication across time and space, as for example, a newspaper) and changing language from the oral to the visual field. The main differences between speech and writing seem to be the fact that they have designs with different objectives: a more permanent (written) and a more transitional (speech) duration. The differences in form between written and spoken language are: the syntax of spoken language is typically less structured than writing, it has incomplete sentences, and typically has little subordination; in conversation normally active forms are used; there are no subject references when mentioning something in the environment and the same syntactic form may be repeat many times. Written language has rhetorical concerns with the organization, greater use of markers of metalanguage (but, when...) and allows displacement of the subject. In the analysis of discourse, grammarians often concentrate on a particular data set and produce a complete set of economic rules to identify a set of sentences illustrating the particular type of text or phenomena being studied. Discourse analysis is typically based on linguistic response from someone (other than the analyst), with two main approaches. The first, analysis with restricted data, is an approach found in much of the linguistic

following Chomskyan tradition where great importance is given to a set of fixed written rules of grammar that define a notion of grammatical vs ungrammatical language. The second, the general data analysis, has its starting point at discursive practices which use a different aspect of a rule-based language, allowing the discussion of regularities because the data exemplify non-categorical phenomena. Givón (GIVÓN, 1979) claims that answering the question “What is the difference between a rule of 90% fidelity and one of 100%?” is next to nothing in terms of psycholinguistics, but in communication, a system with 90% fidelity is a highly efficient system.

In the literature of discourse analysis there are producers and receivers of discourse, but the analysis focuses only on the producer (words-on-the-page). In the approach of Cohesion (text as a product), the sentences together form a text as the words form a sentence, and the anaphoric elements are used to facilitate understanding. In the approach to discourse as a process words, phrases and sentences are considered as evidence of the message being communicated. Doing discourse analysis involves analyzing syntax and semantics, but consists mainly of making a pragmatic analysis to identify what the users of language are doing and the meaning of linguistic features in speech as the means to what they are doing.

The Natural Language Understanding (NLU) by computers began in 1950 as a discipline related to linguistics. This evolved to incorporate aspects of many other disciplines (such as artificial intelligence and lexicography) (BATES, 1994). For Allen (ALLEN, 1995) a good way to define NLU is to consider different applications that the researches address. These applications can be divided into two broad classes: (1) text-based applications and (2) dialogue-based applications.

1. Text-based applications involve the processing of written text, such as books, journals, reports, manuals and more. This class of applications are systems focused on finding appropriate information, information extraction, machine translation and automatic summaries.
2. Dialogue-based applications involve human-machine communication. Typically these applications include systems like questions and answers, personal service by phone and automated tutoring.

Allen also describes the following forms of knowledge relevant to NLU (ALLEN, 1995):

- Phonetic and phonological knowledge: focus on how words are related to the sounds they produce;
- Morphological knowledge: focus on how words are constructed by basic units of meaning

(morphemes);

- Syntactic knowledge: focus on how words can be put together to form correct sentences and to determine the structural role of holding all the words in the sentence and what phrases are subparts of other phrases;
- Semantic knowledge: focus on what words mean and how these meanings combine to form other meanings;
- Pragmatic knowledge: focus on how sentences are used in different situations and how to make interpretations of the sentences;
- Discursive knowledge: focus on how the preceding sentence affects the next; and
- World knowledge: includes general knowledge about the structures of the world that the user of language must have in order to hold a conversation, for example.

Bates (BATES, 1994) observes three general types of widely used semantic representation: propositional logic (commonly semantic representation based on equivalent frames, since it does not allow quantification); first-order predicate logic (FOPL), and several representations that can deal with the expressions that FOPL can not represent.

2.1.2 Approaches to Computational Understanding

NLU could be considered as the translation from a natural language to a computer internal representation, where this representation is computationally easier to process than natural language. There are several approaches for this translation and according to Bates (BATES, 1994) the main ones are statistical, pattern matching, syntactically driven parsing, semantic grammars and case frame instantiation.

- **Statistical based:** The basic proposal of the statistical approach is that the terms occurring in similar contexts carry semantic information similarly. Thus approaches such as Latent Semantic Analysis (LSA), Naive Bayes and Markov models calculate the co-occurrence of terms in texts. A common problem with the statistical approaches is the limited model generated, for example, to detect cases not provided for training (POPESCU, 2005);
- **Pattern Matching:** To Bates (BATES, 1990) the phrase pattern matching approach for the analysis of natural language is based on the interpretation of such phrases as a whole instead of building its interpretation through the combination of structures and meanings

of words or other constituents. In this approach the interpretation is obtained by pattern matching from words. Associated with each pattern is an interpretation, and in the simplest case, this arrangement is simply a list of equivalence classes of expressions and interpretations. Examples of this approach are ELIZA (WEIZENBAUM, 1966) and ALICE (WALLACE, 2011). In more sophisticated variations of this approach, the matching may involve higher-level components or elements of semantics (as labeling by ontologies and entity recognition) so, some aspects of interpretation can be constructed but the approach parameters remain as directly linked as possible to the input;

- **Syntactically driven parsing:** The syntax provides ways of combining words to form higher level units such as phrases and sentences. Syntactically driven parsing is naturally constructive so that, for example, the interpretation of a large group of words is built from the interpretations of its syntactic parts. In this sense, it is the opposite of the pattern matching. The usual way for it to operate is by building a complete syntactic analysis of the phrase and then build an internal representation. This leads to considerable inefficiency, and it is used by approaches to mix analysis and interpretation (BATES, 1990);
- **Semantic grammars:** Analysis of language based on semantic grammars is similar to the syntax-driven analysis, except that it allows for semantic as well as syntactic definitions. Thus the category “noun phrase” in a syntactic grammar would have an additional semantic specification. Semantic grammars are useful mainly to applied NLP, but not for general NLP (BATES, 1990). The two major classes of problems with this approach is that the size of the grammar grows proportionally with the number of knowledge domains and syntactic patterns involved, and most grammars developed for specific fields cannot be extended to new areas (POPESCU, 2005); and
- **Case frame instantiation:** It consists of key concepts (head concepts) and a set of roles (secondary concepts) associated in a well-defined way to the main concept. Initially, the head consists of a main verb and the case includes the “agent” (which performs the action), the object (which suffers the action), location (where the action takes place) and so on.

In this work we adopt the use of semantic grammars, using the Minimal Recursion Semantics (MRS) formalism for producing logical forms for sentences in natural language. The use of logical forms in the NLU part of conversational agents is a solution to the process of converting natural language into a formal model (an important element for the remaining steps). Minimal recursion semantics (MRS) in this sense is a framework for computational semantics that is suitable for parsing and generation and that can be implemented in typed feature

structure formalisms. MRS enables a simple formulation of grammatical constraints on lexical and phrasal semantics, including the principles of semantic composition. This is an approach for semantic representation of large-scale linguistically-motivated computational grammars of natural language. It is based on the belief that grammars should support both parsing and generation, and should be useful for multiple applications, including natural language interfaces of various sorts and machine translation. Our main general criteria for computational semantics are:

- Expressive Adequacy: The framework must allow linguistic meanings to be expressed correctly;
- Grammatical Compatibility: Semantic representations must be linked cleanly to other kinds of grammatical information (most notably syntax);
- Computational Tractability: It must be possible to process meanings and to check semantic equivalence efficiently and to express relationships between semantic representations straightforwardly; and
- Underspecifiability: Semantic representations should allow underspecification (leaving semantic distinctions unresolved), in such a way as to allow flexible, monotonic resolution of such partial semantic representations.

The assumption behind MRS is that the primary units of interest for computational semantics are elementary predications or eps, where by ep we mean a single relation with its associated arguments (for instance, *beyond(x, y)*). In general, an ep will correspond to a single lexeme. MRS is a syntactically 'flat' representation, since the eps are never embedded within one another. An alternative is to modify the form of the semantic representation, in particular to use a non-recursive, or flat representation such as those developed by (PHILLIPS, 1993) or (TRUJILLO, 1995).

In this work we are interested in the use of intentional information, and two systems that use it are Phoenix and Why2-atlas. The Phoenix system (WARD; ISSAR, 1994) is designed for robust information extraction, using a simple mechanism to represent a task frame semantics. The system uses Recursive Transition Networks for encoding semantic grammars. Grammars specify patterns of words which correspond to semantic tokens of system understanding. A subset of tokens is considered as high-level, which can be recognized regardless of context. Networks are calling other networks to produce a tree of semantic analysis. The high-level tokens appear as slots in the frame structure. The frames serve to associate a set of tokens with a semantic function. The output of the analysis is the frame name and the tree with filled slots.

The project Why2-atlas (VANLEHN et al., 2002) had three objectives: build and qualitatively evaluate a physics tutor, where all students communicate using natural language (textual), compare various techniques of NLP, and develop an authoring tool that facilitates the development of tutoring systems based on NL. Why2-atlas is composed of several modules: the sentence-level understanding (SLU), the discourse-level understanding (DLU), the tutoring strategies and the dialogue engine. These four modules are controlled by the discourse manager. The language understanding (SLU) converts each sentence into a set of propositions of first-order logic. The SLU is composed of a parser, a module repair (spelling) and a statistical analyzer. The level of understanding of discourse (DLU) receives a logical form and provides as output a logical proof. This test is constructed with Tacitus-lite+ which is an extension of Tacitus. The knowledge base is a set of clauses that represent correct beliefs about physics.

2.2 Dialogue

This section explains the elements of dialogue and conversational agents discussing the forms of implementation of dialogue systems (both chatterbots systems and more complex systems) and plans to represent the conversation.

2.2.1 The Dialogue Features

Human conversation is an intricate and complex joint activity. Because of the limitations of our current technology human-machine conversations are simpler and more constrained than human conversations. Nonetheless we attempt to design a conversational agent to talk with humans, it is crucial to understand something about how humans speak with each other. Some properties of human conversation (that distinguish it from the kinds of (text-based) discourses) are turns, speech acts, grounding, conversational structure and their implications. We show in Figure 2.1 (from (LEE et al., 2008)) the components of an architecture based on several virtual humans. This architecture has 3 main components: the environment represents the agent interface with the world; the body is responsible for the agent's personification; and the mind manages the mental aspects. In this work we focus on the mind of an agent assuming that the communication with the environment is performed via text. The agent body is out of the scope of this work.

According to Lester (LESTER; BRANTING; MOTT, 2004) the accuracy and efficiency

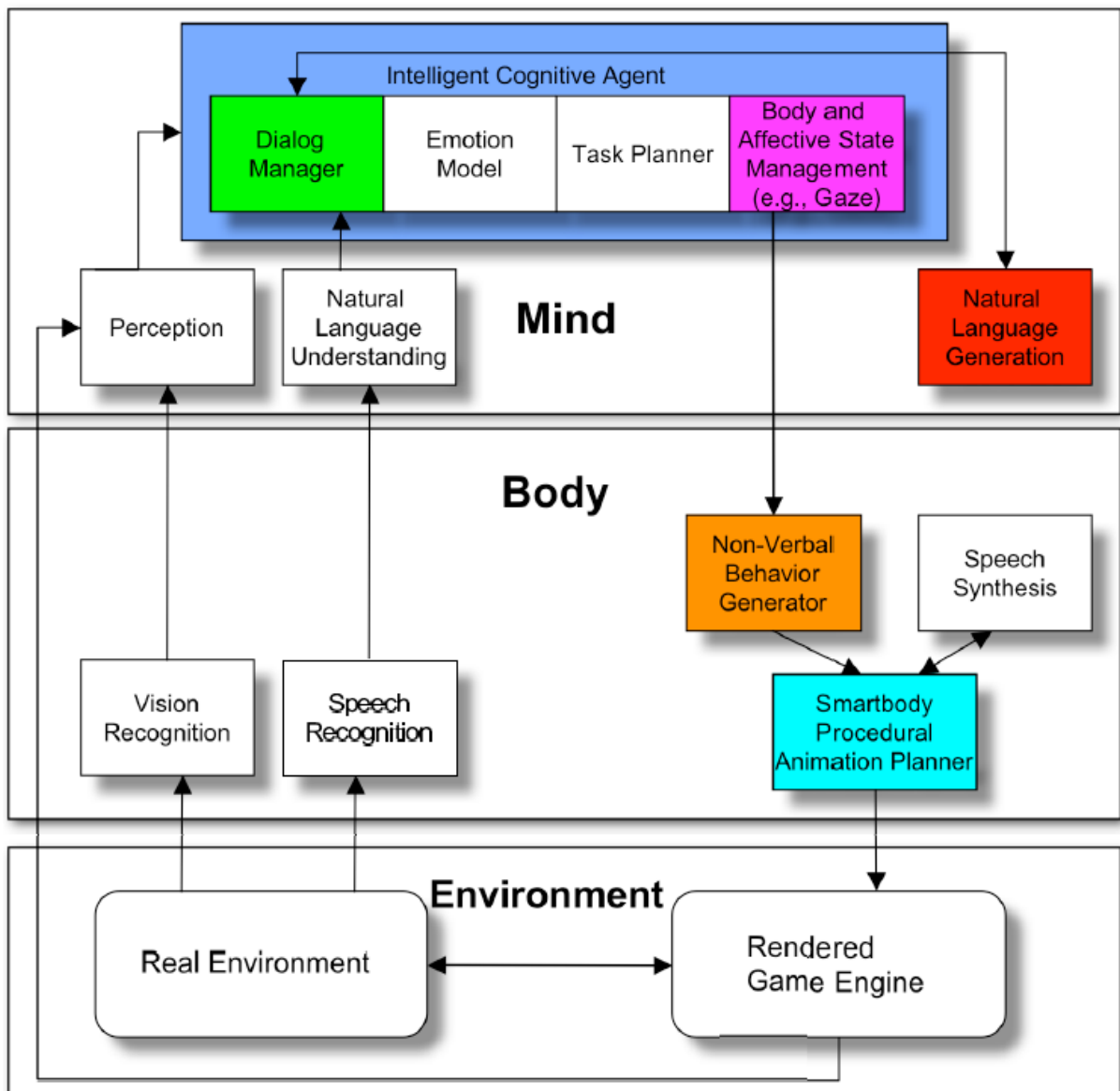


Figura 2.1 – The virtual human system architecture.
 Fonte: (LEE et al., 2008)

in natural language processing are essential to an effective conversational agent. To answer one sentence (question, statement or order) of the user an agent must perform three steps: (1) interpret the phrase, (2) determine what actions should be taken in response to the statement and (3) perform actions, which may include responding with texts from the web or other sources to perform system actions (for example, recording information). The interpretation of the sentence is transmitted to the dialogue manager module, which is responsible for determining a response action. Appropriate actions depend on the interpretation of the user sentences and the dialogue state that represents the current goals of the agent in conversation. The new state of dialogue is generally a function of the current state of the user's sentence and the information available. The response generator module has two categories of responses: communication with the user (such as texts, websites, emails, images or other forms of communication) and non-communicative responses (such as updating the user profile).

Recently, with the objective of defining representation languages that may serve as clear interfaces at different levels of abstraction needed to modularize the problem. In this sense the framework SAIBA (acronym for Situation, Agent, Intention and Behavior and Animation) has the human-like behavior as a major contributing factor. In this framework the three major stages of processing correspond to modules: (1) planning of a communicative intent, (2) planning a multimodal behavior which can carry out these intentions, and (3) performance of planned behavior (RUTTKAY, 2008). The interfaces between the two levels are provided by two markup languages: Function Markup Language (FML) between levels 1 and 2, and Behavior Markup Language (BML) between levels 2 and 3.

In an interaction, a turn indicates a change of speaker, and, generally, there is only one participant speaking at a time, and if there is any overlap, it is easily solved. The order and distribution of the shifts are not fixed but varied, and neither are the size and length of the turn. A turn can consist of a simple lexical item (word), phrases, clauses or complete sentences.

When someone says something, it is assumed that their speech is pertinent, relevant to the immediately preceding sentence, indicating the idea that sentences are linked in pair rules. Adjacency pairs need not to be strictly adjacent, for example, inside a pair there can be other pairs required to complete a sub-goal (that have to be solved in order to do the top-level task).

It is also important to determine whether information in a discourse goes beyond what is literally expressed in the individual utterances, and most research on this topic falls into informational and intentional categories. The informational approach states that the coherence of the speech is due to semantic relations between the information carried by successive utterances. The intentional approach states that the coherence of a discourse derives from the intentions of

speakers and writers, and understanding depends on the recognition of these intentions, and the early works in this area are based on the speech acts theory (SEARLE, 1975).

Current approaches to discourse and dialogue use more than intentional and informational view, they take into account elements like:

- Set of coherence relations, recursively applied to segments of discourse.
- The focus of the speaker's attention, the sequential and intentional structure of utterances (language structure).
- A hierarchical organization of the text parts as core (central) or satellite (support) of one of a set of discourse relations. This approach is known as Rhetorical Structure Theory (RST).

Austin (L., 1962) presents the concept of a sentence in a dialogue as a kind of action, or speech acts, performed by a speaker. He considers that any sentence includes three types of acts: (1) a locutionary act is the production of an utterance, with syntactic and semantic aspects; (2) an illocutionary act, with the real, intended meaning of the utterance, and (3) a perlocutionary act, with the effects of the utterance itself, on the feelings, thoughts or actions. The term speech act is used to describe illocutionary acts more frequently than the other acts. Searle (SEARLE, 1975) created a taxonomy with five major categories (assertives, directives, commissive, expressive and declarative), these are shown in Table 2.1. With the idea of agents undertaking actions as a response to evidence (grounding), Clark (CLARK; SCHAEFER, 1989) introduced the idea of contribution (or linguistic act), composed of presentation (speaker provides information) and acceptance (listener decides to accept using their knowledge and beliefs).

assertive	suggest	I suggest that you think about this.
directive	advise	I advise that you think about this.
commissive	plan	I plan to think about this.
expressive	thank	I would like to thank you.
declarative	dismiss	You are dismissed of doing so.

Tabela 2.1 – Searly taxonomy

According to Jurafsky and Martin (JURAFSJY; MARTIN, 2009) the approaches for modeling dialogue interpretation that stand out are dialogue grammars and plan-based models. Dialogue grammars are based on the observation that there are a number of sequences in regular dialogue. Rules state restrictions on acceptable dialogues as well as grammatically acceptable word chains. The elements of these rules are typically illocutionary and describe stages of dialogue. Plan-based models are based on the fact that people do not realize random conversations, but plan actions to accomplish their goals. This model has limitations on the recognition

of the illocutionary act, but it is nonetheless able to solve problems of interpretation through non-linguistic methods, for example, plan recognition (USZKOREIT, 1997).

Jurafsky and Martin (JURAFSKY; MARTIN, 2009) presents three techniques commonly applied to managing computational dialogue. (1) Finite state automaton (FSA) where transitions correspond to individual statements and states correspond to the goals of the agent. FSA is suitable for simple dialogues in which all possible sequences of states of the dialogue can be anticipated. (2) Algorithms for interpretation of form, since FSA is not suitable for dialogues in which the order of statements is unpredictable, because the user can provide several pieces of information in a simple sentence. (3) Joint initiatives occur when a person answers a question with another question, as when additional information is needed to answer the question. The Figure 2.2 illustrates the three techniques, it models a travel dialogue manager (it is a simplification of a domain in which the user needs to inform the travel date, origin city and destination city). In this figure, model 1 (FSA) requires the information in a specific order; model 2 accepts the unsorted information; and model 3 considers that the user does not know the destination cities, hence there is a sub-dialogue to provide more information.

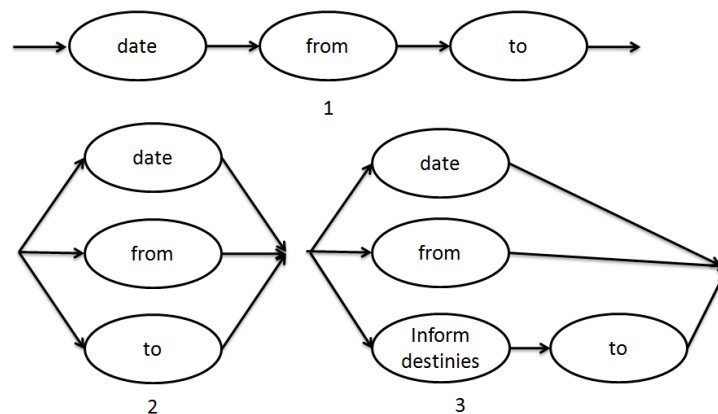


Figura 2.2 – Illustration of the three techniques to managing computational dialogue

The industry has reached a maturity in dialogue characterized by a vertical structure of technology providers, platform integrators, application developers and hosting companies. At the same time industrial standards are pervading the underlying technology and providing higher levels of interoperability. On the one hand commercial dialogue systems are largely based on a pragmatic approach which aims at usability and task completion. On the other hand spoken dialogue research has focused on a parallel path trying to attain naturalness and freedom of communication (PIERACCINI; HUERTA, 2005).

2.2.2 Chatterbots

A chatterbot is a computer program designed to simulate an intelligent conversation with one or more human users via auditory or textual methods. One of the first implementations of conversational dialogue is ELIZA (WEIZENBAUM, 1966). As an evolution of ELIZA, there is ALICE (WALLACE, 2011), that presents a conversation that is easier to control and to develop. The development of a ALICE like chatterbot is by Artificial Intelligence Markup Language (AIML) and in this work we opted for the use of an ALICE chatterbot as a simplification of the communicative acts. Both, ELIZA and ALICE systems use the FSA approach and have complex knowledge bases to do a complete conversational structure.

Alice uses Case-Based Reasoning (CBR), according to which the algorithm searches the pattern that best fits for each input. The system searches for the category to match the user input then answers using the template information. Conceptually, Alice is not different from Eliza. The main difference lies in the amount of base cases and in the tool that creates new content dialogues by analyzing the previews dialogues.

AIML describes a class of data objects called AIML that is derived from XML (eXtensible Markup Language). AIML was developed by Dr. Richard Wallace and the free software community Alicebot. This formed the basis of the first Alicebot, ALICE (Artificial Linguistic Internet Computer Entity). The objectives in the design of AIML (WALLACE, 2011) are:

- AIML should be easy for people to learn;
- AIML should codify the minimum concept necessary to model a knowledge of stimulus-response, as the original ALICE;
- AIML should be compatible with XML;
- AIML objects should be legible and reasonably clear to people; and
- AIML should not contain any language dependencies.

Each AIML object has a logical and a physical structure. A physical structure of the object is composed of units (topics and categories). The object can be in the root or in an entity. A logical structure of the object is composed of elements and references, which are indicated by marking the object explicitly.

AIML objects are composed of topics and categories that contain data or AIML elements. The AIML elements encapsulate the knowledge of stimulus-response contained in a document. The data are elements that can be interpreted by the AIML interpreter or treated later as an answer. The following AIML example illustrates two greeting categories where the

first answers with the template “*Hi there!*” when the input pattern is “*HELLO*” and the second for the input matching “*HI*” forwards to pattern “*HELLO*” (because of the logical tag “*srai*” that associate one data with other).

```
<category>
<pattern>HELLO</pattern>
<template>Hi there!</template>
</category>
```

```
<category>
<pattern>HI</pattern>
<template><srai>HELLO</srai></template>
</category>
```

2.2.3 Conversational Systems

A conversational system can be viewed as a system able to interact with user using natural language. Examples of conversational systems, such as GALATEA and Rea, will be discussed in this section, as they will serve as the basis for the relation between NLU and dialogue manager in this work.

Rea (CASSEL et al., 2000) has a fully articulated body, 3D graphics and communicates with verbal and nonverbal modalities. It is capable of fully describing a house using a combination of speech and gestures, and can also respond to input from the users (verbal and nonverbal). When the user makes signals typically associated with a turn-taking behavior such as gesturing, Rea allows itself to be interrupted, then takes the turn again when it is able. It is able to initiate conversational repair when it does not understand what the user says. Rea speech and gesture output is generated in real time from the same knowledge base and a description of the basic goals of communication. Two cameras mounted on top of the projection screen track users’ head and hand positions in space. Players use a microphone to communicate speech input.

GALATEA (SKANTZE, 2005) is a module for conversational spoken language where sentences are interpreted in context. The recognition result of the Automatic Speech Recogni-

tion (ASR) is sent to the interpreter, called PICKERING, which recognizes and creates semantic representations of communicative acts of the user. The communicative acts are sent from PICKERING to GALATEA which creates a context and builds a model of discourse. This model is then sent to the task manager, which consults the discourse model and the base areas to make decisions and send them to the communicative acts system. These acts are sent back to GALATEA and to the language generator. In the generator, the textual representation of the communicative act system is sent to the speech synthesizer. Semantic descriptions are consistently represented as the roots of trees. The nodes in the tree can represent attribute value pairs, objects, relationships and properties. These structures are very flexible and can be used to represent the deep structures of semantics as nested feature structures, or simple shapes, depending on the requirements of the domain. In Figure 2.3, from (SKANTZE, 2005), the semantic representation of the utterance “the building is made of wood” used by GALATEA is shown.

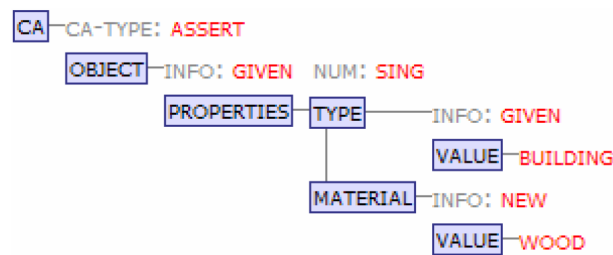


Figura 2.3 – The semantic representation of the utterance “the building is made of wood”.

Nakano and colleagues (NAKANO M., 2008) present a framework for building control modules on a symbolic level of animated agents and robots with a spoken dialogue interface. This has distributed modules called experts, each of which is specialized in performing certain types of tasks. The main features of this framework are:

1. multi-domain dialogue: the agents are expected to perform various tasks, so they must work in various areas and change from one domain to another in response to user’s statements;
2. interruption handling: the ability of dealing with user’s interrupting expressions while speaking or performing tasks is fundamental to the human interaction agent;
3. parallel task execution: it is expected to be able to perform multiple parallel tasks when possible; and
4. extensibility: how agents can be used for a variety of tasks, it should be possible to incorporate many strategies for dialogue and planning tasks.

It also presents a framework called RIME (Robot Intelligence based on Multiple Experts), which uses modules called specialists. Its basic idea is to specify a common interface

expert to set them up and get flexible control. The modules are:

- understander: which is responsible for speech recognition;
- action selector: which is responsible for the selection of actions; and
- task planner: which is responsible for deciding which specialist should work to accomplish tasks.

2.2.4 Beliefs, Desires and Intentions (BDI)

The BDI is a widely used model mainly because of the level of abstraction required of modeling complex behavior. Of the seven aspects of intelligent behavior (perception, planning, commitment, acting, beliefs, desires, intentions) four are processes and three are part of the agent's cognitive states (beliefs, desires and intentions). These states can be used with plans for generating and interpreting sentences, since these plans are desires that merged with beliefs turn into intentions to be performed by agent. Using plans to generate and interpret sentences in this way requires that the planner have good models of its beliefs, desires, intentions (BDI), as well as those of the interlocutor. Plan-based models of dialogue are those often referred to as BDI models. BDI models of dialogue were, probably, first introduced by Allen, Cohen, Perrault (COHEN; PERRAULT, 1979).

BDI is a cognitive architecture to intelligent agents in the model of human practical reasoning (BRATMAN; ISRAEL; POLLACK, 1988) with three mental states: belief, desire and intention. A BDI architecture represents the internal processes through the mental states mentioned above, and defines a control that selects a rational course of action.

The main idea is that the cognitive agent has internal states which are related to its environment. These states would correspond to human mental states, which make a link with the world in terms of existence and significance.

Beliefs represent knowledge about the world that the agent has (WOOLDRIDGE, 2000), which may even be incomplete or incorrect. From a computational point of view, beliefs are only one way to represent the world (either through variables, a relational database, or symbolic expressions in a predicate calculus). Beliefs are essential because the world is dynamic, and the systems have only a local view of the world (events outside its sphere of perception should be remembered) (RAO; GEORGEFF, 1995).

Desires are any related world states that the agent wants to provoke. However, even if an agent has a desire it does not mean that it should act to satisfy it. That means that before

deciding what to do an agent goes through a process of rationalization and confronts his desires with its beliefs. The agent will adopt a desire that is possible according to some criterion. The desires are an essential component of the system as they represent an end state that the agent wants to check (RAO; GEORGEFF, 1995).

Intentions could be considered a subset of desires, but unlike the latter, the former should be consistent. Intentions are formed from a deliberation process and from the refinement of other intentions by the agent and can also be entered by the user. Usually the term intention is used to characterize both mental states as actions. The intention mental state is directed to the future and it not necessarily triggers an action. The intentional action is directed to the present and represents the act of taking action immediately (BRATMAN; ISRAEL; POLLACK, 1988).

The implementation of a BDI agent can be done with AgentSpeak(L) using the Jason system. The programming language AgentSpeak(L) (RAO, 1996) is one of the extensions of logic programming for the BDI agent architecture, it provides a framework for programming BDI agents. An agent is created by specifying a set of beliefs and a set of plans. It distinguishes two types of objectives: achievement goals and test goals. Achievement goals assert that the agent intends to achieve a state of the world where the associated predicate is true. A test goal returns a unification for the associated predicate with one of the agent's beliefs.

Plans refer to the basic actions that an agent is able to perform in its environment. Such actions are defined the same way as first order predicates, but with special symbols used to distinguish them. A plan is formed by a triggering event, followed by a conjunction of literals representing a context beliefs. The context must be a logical consequence of the basic beliefs of the agent for the plan would apply. The remainder of the plan is a sequence of basic actions or (sub)-goals that the agent has to achieve (or test), after the plan is chosen for execution.

A triggering event defines which events may start executing a plan. An event can be internal, when a subgoal has to be achieved, or external, when generated from belief updates as a result of perceiving the environment. There are two types of triggering events: those related to the addition and removal of mental attitudes (beliefs or goals).

In each interpretation cycle of the agent, the AgentSpeak(L) updates the list of events, which can be generated from the perception of the environment or the implementation of intentions (when subgoals are specified in the body of plans). It is assumed that beliefs are updated from perception and whenever there are changes in the agent's beliefs, which implies the inclusion of an event in the set of events.

2.3 Summary

This section presented methods for understanding natural language (translating natural language to a controlled language), dialogue control (methods of retaining structure and coherence) and systems to make information available.

In NLU we showed the forms of knowledge (phonetic, phonological, morphological, syntactic, semantic, pragmatic, discursive and world) and the approaches to computational understanding (statistical based, pattern matching, syntactically driven parsing, semantic grammars and case frame instantiation). Our main focus was on translating natural language into a logical language (predicate logic). In addition to the advantages of semantic representation, we also obtained a form compatible with BDI system used in this work, thus enabling a clear interface between the modules of NLU and dialogue manager.

3 QUESTION ANSWERING

The task of a Question Answering (QA) system is to automatically answer a question in natural language, searching for information in a given data source, such as structured databases or unstructured natural language documents (e.g. corpora from a given domain, newspaper texts). This is a challenging task as question types are varied and can include facts, lists, definitions, while answers may come from sources of information ranging from small document collections to the World Wide Web, and these determine the level of processing that can be feasibly employed for analyzing the question and data sources (JURAFSJY; MARTIN, 2009). Moreover, the difficulty of the task is also influenced by whether the questions are restricted to a particular domain (e.g. sports, genes) or not, which additional sources of information are available for a given language (e.g. ontology of domain-specific knowledge, general ontology's), their coverage, and which tools can be used to help the task (e.g. named entity recognizers', parsers, word sense disambiguation tools). Such systems have the potential to make written information more easily accessible to wider audiences, through questions in natural language, especially if the interface allows voice communication.

The Question Answering process can be commonly divided into four stages, as follows: (1) question analysis, (2) identification of candidate documents, (3) generation of candidate answers and (4) computation of answer score. These stages require the following tasks: (1) identification of the topic of the question; (2) translation of user question to a query for an information retrieval tool and search tool; (3) identification of candidate answers through processing of the documents retrieved; and (4) ranking of the candidate answers in terms of decreasing order of relevance to the question. Figure 3.1 illustrate these four stages using as example sentence the question "Who is the president of Brazil?".

The relevance of research in QA is demonstrated by the number of initiatives and conferences devoted to the task. For example, several conferences have had special tracks for evaluating Question Answering technology and systems, such as Text REtrieval Conference (TREC - <http://trec.nist.gov>) and Cross Language Evaluation Forum (CLEF - <http://www.clef-campaign.org>). In addition, the interest in the task is also reflected by the inclusion of languages other than English in these evaluations: since 2004 CLEF has included, among others, Portuguese as one of the languages accepted, both as the language of the questions and as the target language containing the answers, and evaluating both mono and multilingual systems. However, there are still only a few systems for Portuguese, and even less for Brazilian Portuguese. Among the systems taking part in CLEF we can see a development in terms of performance,

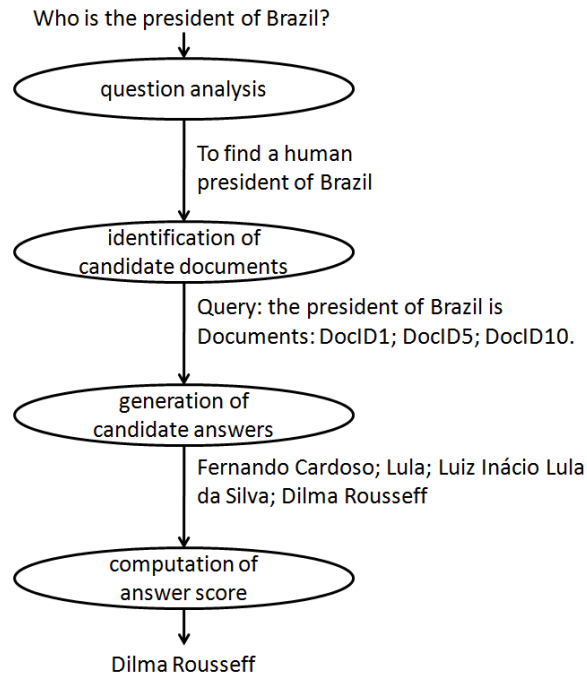


Figura 3.1 – Example of QA stages

as they were evaluated on the same data, over the years. Examples of QA systems for Portuguese are RAPOSA (SARMENTO, 2006), Priberam (AMARAL et al., 2006) and ESFINGE (COSTA, 2004). However, there is no consensus as to the amount of resources and tools that are needed in order to build a working QA system with reasonable performance. One of the sub contribution of this work is to evaluate that and assess in particular the contribution of a parser in this process. In this section we present the main steps in question answering process, which is an interesting approach to identify information not available in the agent's mental model. We then describe an experiment to assess the contribution of a parser in the performance of a QA system.

3.1 The Internal Steps

The process of question answering can be divided into four steps: question analysis, query generation and search, candidate generation and answer scoring.

Question analysis is the first step performed in a QA system, whose goal is to identify the information requested by the user aiming to reduce the number of documents to be scanned. In (POMERANTZ, 2005) several taxonomies of questions are addressed. Among the taxonomies of questions that stands out as being best known and most used is the wh-word or wh-phrase, which considers the pronouns or expressions that indicate the type of information needed to answer the question in English (who, which, what, when, where, why, how). For instance,

Rules	Expected Type
who/whom	PERSON
When	TIME
where/what place	LOCATION
what time day	TIME
what week day	DAY
what/which month	MONTH
what brand	PRODUCT
what	NAME
how far/tall/high	LENGTH
how large/hig/small	AREA
how heavy	WEIGHT
how rich	MONEY
how often	FREQUENCY
how many	NUMBER
how long	LENGTH/DURATION
why/for what	REASON

Tabela 3.1 – Rules for identifying the type of response expected
 Fonte: (SRIHARI; LI, 2000)

Harabagiu (HARABAGIU; MAIORANO; PASCA, 2003) analyses several sets of rules, some are listed in Table 3.1, to which the question refers to using only the wh-word. In (SRIHARI; LI, 2000) 16 rules are presented based on pairs of wh-word complement indicating what kind of object the question refers.

Query generation aims to translate the user question to a query for an information retrieval tool. The largest division of work in this stage refers to the system domain, which may be open or closed. Systems often focus on a closed domain containing local collection of documents on a specific subject. In this case traditional information retrieval tools are used to identify the most relevant documents to the query. Open domain systems use dynamic collections, which are usually based on web pages due to the large amount of information available on the web. Document retrieval is done through search engines like Google, Yahoo and Wikipedia, and documents indicated as most important are analyzed. The main steps consist of preparing the documents for the identification of relevant terms, standardization of terms, expansion of query terms and query formatting. These steps are performed with the goal of identifying the most relevant documents for the information retrieval tool. The main technique used in QA for the identification of candidate answers is the conversion of documents into vectors and comparing them in terms of the relevant terms. The comparison between documents is performed by calculating the cosine between them, and it has the greatest value when two vectors have similar components. This technique is widely adopted because of its efficient performance even

with large amounts of data. The conversion of documents into vectors can use different granularities for representing the user sentence. The granularity is a factor that has high impact on the system's quality, because a low granularity, such as a word or phrase, causes increased processing (since there are many vectors) and may result in loss of information (because the information could be larger than a phrase), and a document with high granularity requires the later identification of the relevant part of the document where the answer lies.

The extraction of information from documents for the creation of vectors can be done in several ways, the most common are using bag of words, n-grams and markups. The bag of words technique simply separates the document into words and each word consists of a dimension in which the vector is represented. The n-grams approach consists of not only using words individually as vector dimension, like in bag of words approach but also of using n adjacent. Finally, a document can be preprocessed in order to do a markup annotation, and the vectors' dimensions are the markings themselves. These three approaches are exemplified in Figure 3.2.

Sentence:	The president of Brazil is Dilma Rousseff
bag of words	["the"; "president"; "of"; "brazil"; "is"; "dilma"; "rousseff"]
n-gram:	["the president"; "president of"; "of brazil"; "brazil is"; "is dilma"; "dilma rousseff"]
markup:	[title:president; country:brazil; name:dilma; name:rousseff]

Figura 3.2 – Example of extraction of information approaches

The evaluation is often based on simply counting the occurrence of a term in the document. This technique poses some problems with very common words in documents and rare in the question, such as stopwords (like “the” and “of”), and tends to privilege short documents because they have a higher proportion of similar terms to the question. Due to the low cost of this technique it can be used in a binary version, which only identifies the presence or absence of the word in the document, thus not depending on the size of the document. The use of the n most frequent words is another option where only few words in the document are considered, so as to try to use only the terms which the document addresses.

The use of filtering of terms makes use of heuristics acquired in the first step (question analysis) such as what type of object or what terms the answer should include. With these heuristics all parts of the returned document that does not satisfy the heuristics can be ruled out, thus getting only the relevant parts of the document. For reasons of efficiency or speed, it is possible to reduce the set of answers returned, for example, adopting the identification of only one answer, by increasing the restrictions imposed by the heuristics until only one answer is returned.

3.2 Question Answering Systems

In this section we discuss several QA systems implemented subsequently focusing on those developed for Portuguese.

QuALiM (KAISSER, 2005) explores the use of lexical resources like FrameNet, Prop-Bank and VerbNet in QA and considers two different complementary methods that use these resources. One method uses data from these resources to generate potential answers containing the exact user sentence searched on the Web. The second method uses only a search based on keywords, which may generate a large set of candidate sentences.

TrueKnowledge is a knowledge-based system, in other words, a large database of facts about general topics. Facts are also implied by a knowledge generator or an external source.

LAMP (LIN, 2005) has the special characteristic of using only the snippets returned by Google search engine. This work assumes that the returned snippets provide an important practical characteristic for online QA systems, due to the low cost of downloading and analyzing in comparison to the original web documents. The score of LAMP is not as high as the best QA system in TREC, this discrepancy is due to several reasons, among them the fact that the answer is evaluated with a regular expression in TREC and the fact that many correct answers are judged as deficient for not being specified in the TREC collection of documents or for changing over time. A problem of this system is the time spent on analyzing the answer, due to the use of the Web.

Ephyra or OpenEphyra ¹ is the first open framework for QA which recovers accurate answers to questions in natural language using Web sources. The framework provides implementations of algorithms proved effective in TREC. In the pipeline provided Ephyra's answer extraction and selection stages have been combined. This takes into account that both phases perform similar operations on the same structures and data. The system performs a syntactic and semantic analysis and a query to a search engine is generated and performed. WordNet is used to expand query terms with semantic concepts. Ephyra uses 154 types of answer organized in a hierarchy with 44 top categories.

AnswerBus QA system (ZHENG, 2002) is an open field of information retrieval based on sentence level. It accepts English, German, French, Spanish, Italian and Portuguese and extracts possible answers from the web using the search engines Google, Yahoo, WiseNut, AltaVista and Yahoo News as possible bases for answers. It was tested on TREC-8 with 200

¹OpenEphyra is available for download on SourceForge under the GNU General Public License (GLP): <http://sourceforge.net/projects/openephyra>

questions and obtained 70.5% accuracy. A simple module for recognition of language determines the language of the question and if it is not in English, it is translated using the tool of the Alta Vista Babel Fish.

3.3 Question Answering for Portuguese

The approach of the University of Evora (QUARESMA et al., 2004a) is based on selecting for each question the most relevant documents, following an information retrieval task. The information contained in each document is analyzed and the query is processed over the knowledge base of each text for the system to return the first answer found. To create the knowledge base for each document, the parser PALAVRAS (BICK, 2000) was used, and the parsed sentences are rewritten using first-order logic. The system uses an ontology built on the basis of the first order logic expressions. For answer identification, the same processing is done to analyze both the question and the documents.

The Priberam system uses a internal system called TRUST (Text Retrieval Using Semantic Technologies) for information retrieval combined with deep linguistic analysis such as named entity recognition, part-of-speech (POS) tagging and context-free grammar for disambiguation, employing rich resources like ontologies and thesauri (AMARAL et al., 2006). It follows 5 stages: (1) indexing, (2) question analysis, (3) document retrieval, (4) passage retrieval and (5) answer selection. The analysis of the question determines the type and topic of question, and these are used as basis for document retrieval. Passage retrieval is done on the basis of similarities between a passage and the question, prioritizing sentences with more complete information (e.g. *Fidel Castro vs Fidel*). To extract an answer, the system determines the similarity between each candidate answer and the types identified during question analysis.

RAPOSA (SARMENTO, 2006) is an open domain QA system which adopts a shallow approach for answering biographical questions. Questions are divided into 3 groups: elementary, profile-dependent and speculative questions. The first type refers to questions about common and general attributes (e.g. parenthood), the second depends on the particular person and requires world knowledge for a precise answer (e.g. *name one movie directed by Claude Lanzmann*), while the third may involve a false hypothesis (e.g. *when has X committed suicide?*). For a given question the system performs the following steps: (1) identification of named entities; (2) query generation from the question; (3) collecting of summaries where the answer may be found, (4) extraction of answer using a set of rules and a measure of the semantic similarity between the question and the possible answers, (5) merging of answers and (6) answer

selection based on a confidence score of a summary given the question. During the analysis of the question the system identifies the type of question and reformulates it so that it only contains core words. The system then searches both a document collection and the Wikipedia and returns summaries of the texts. The extraction of the answer is done selecting the documents whose summaries have the same semantic type as that of the question, according to a set of rules (SARMENTO, 2006). In order to assess the effects of the semantic analysis of texts on a realistic application an evaluation of the system was performed using the CLEF 2007 dataset, and it resulted in around 20% incorrect answers. These errors may be partly explained as caused by the search in summaries and by the choice of summaries with a semantic permissible type.

Esfinge (COSTA, 2004) performs question reformulation and n-gram harvesting, filtering and composition. The question reformulation module identifies patterns based on the words in the question and rewrites the question to turn it into the expected form for the answer. The n-gram harvesting module uses the rewritten form obtained in the previous module to query the document repository for the highest scoring ngrams in the summaries. The n-gram filtering module uses POS information about n-grams for reducing the candidate set, and these are given to the n-gram composition module to try to answer the question, and it determines characteristics of the answer (e.g. whether it should be in the singular or plural).

In 2004 two systems took part in the CLEF evaluation: the University of Évora (QUARESMA et al., 2004a) system and Sphinx. The former had the best performance with 28.64% overall accuracy, where 29.17% were obtained for factoids and 25.81% for definitions, while Sphinx with web access had the second best (QUARESMA et al., 2004b). In 2005 Priberam also took part, and had the best performance (with 64.5% accuracy, where 67.41% were for factoids and 64.29% for definitions), while the University of Évora system obtained 26% accuracy, where 21.48% were for factoids and 35.71% for definitions (VALLIN et al., 2005). There was an increase in the number of collections in 2006, which included the Portuguese newspapers Público and Folha de São Paulo. From the four participating systems (Esfinge, NILC, Priberam and Fox), Priberam had the best performance (with 67% accuracy) while Sphinx again came second (with 25%) (MAGNINI et al., 2006). In 2007 5 systems took part in CLEF: University of Évora, Sphinx, INESC, Priberam and Fox. Priberam had again the best overall performance (50%) but this time there was a smaller gap to the second best performance, University of Évora (42%) (GIAMPICCOLO et al., 2007). In 2008 the pattern was repeated, and among the 6 participants (University of Évora, Esfinge, Fox, INESC, Priberam and Universidade Aberta), Priberam obtained the best performance with the University of Évora system coming second (FORNER et al., 2008). Figure 3.3 summarize the performance of the Portuguese language in

each of the years of evaluation by the QA@CLEF.

Given this scenario, although a common evaluation standard helps to compare these systems, their evolution through time and the use of different datasets for CLEF do not allow straightforward conclusions to be drawn about the advantages of different system configurations and the use of particular resources and tools.

3.4 Summary

This work use some resource and tools as basis for the QA system for Portuguese. In particular we use a corpora (Floresta Sintática) and an ontology developed in the Comunica Project the Palavras parse in this work used only as a PoS tagger and the OpenCCG in this work used as a full parser and a translator from natural language form to logical form.

However, in resources-poor languages there are not a common sense about the resources and approaches to use, in this way we present a comparison over resources and approaches in Portuguese QA.

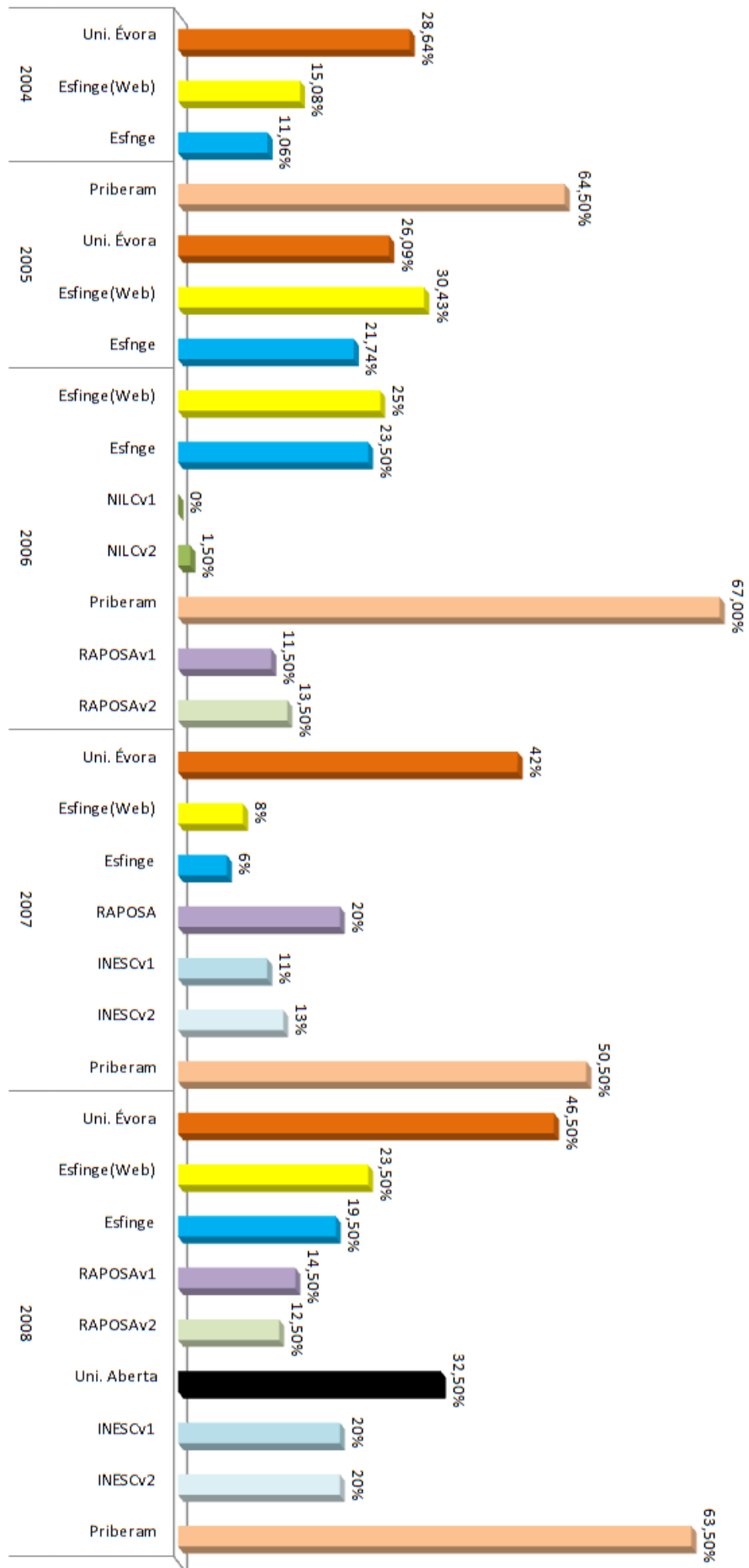


Figura 3.3 – Performance of QA systems for Portuguese, QA@CLEF.

4 RESOURCES AND TOOLS

In this section we present the tools and resources used in this work. The parsing system subsection discusses the system used to make parsing in this work. This section is structured in four subsections. In corpus resources, we describe the corpus used to train and evaluate our parser. The ontology subsection shows the knowledge model used as semantic information in this work.

4.1 Morphosyntactic Information

In order to process a natural language interaction in this work we use PALAVRAS (BICK, 2000) for assigning morpho-syntactic categories to the words in the sentence. These categories will be used for the grammatical acquisition realized in this work.

The parser PALAVRAS (BICK, 2000) is an automatic grammar- and lexicon-based parser for unrestricted Portuguese text. The project combines preceding and ongoing lexicographic work in an effort for automatic grammatical annotation, and has since ventured into higher level syntactic and semantic analysis. “Ultimately the parser is intended for applications like corpora tagging, grammar teaching and machine translation, which all have been made accessible in the form of Internet based prototypes. Grammatical rules are formulated in the Constraint Grammar formalism and focus on robust disambiguation, treating several levels of linguistic analysis in a related manner. In spite of using a highly differentiated tag set, the parser yields correctness rates - for unrestricted and unknown text - of over 99% for morphology (part of speech and inflexion) and about 97% for syntactic function, even when geared to full disambiguation. Among other things, argument structure, dependency relations and subclause function are treated in an innovative way that allows automatic transformation of the primary, “flat” Constraint Grammar based syntactic notation into traditional tree structures. The parser uses valency and semantic class information from the lexicon, and a pilot study on disambiguation on these levels has been conducted, yielding encouraging results” (BICK, 2000). An example of PALAVRAS analysis for the sentence “Tem sentido - aliás, muitíssimo sentido” is shown bellow. Each word in the sentence is shown in a line, and for each word it determines the canonical form in brackets after the word (e.g. [ter]), and part of speech and other morphosyntactic information (e.g. V for verb, PR for preposition, 3S for third person of singular and IND for indicative). “\$” indicates punctuation.

Tem [ter] <mv> V PR 3S IND
 sentido [sentido] N M S
 \$--
 alias [alias] <kc> ADV
 \$,
 muitissimo [muitissimo] <SUP> DET M S
 sentido [sentido] N M S
 \$.

4.2 Combinatory Categorical Grammar (CCG)

In this work, as we need to be able to process unrestricted natural language sentences, we developed a wide-coverage parser automatically induced from corpora. This allow us to process an input sentence and return information such as equivalent logical form which is important for determining the relation between NLU and dialogue manager.

The parser is formalised using Combinatory Categorical Grammar (CCG) (STEEDMAN; BALDRIDGE, 2007) is an efficiently parseable, yet linguistically expressive grammar formalism. It has a transparent interface between surface syntax and semantic representation, including predicate-argument structure, quantification and information structure. CCG relies on combinatory logic, which has the same expressive power as the lambda calculus, but builds its expressions differently. In particular, we are using the implementation of CCG provided by OpenCCG, which is an open source natural language processing library written in Java, which provides parsing and realization services based on Mark Steedman's Category Grammar (CCG). The library makes use of the multi-modal extensions to CCG devised by Jason Baldrige (BALDRIDGE, 2002).

Categorical Grammar (CG) is a lexicalized grammar formalism and Combinatory Categorical Grammar (CCG) (STEEDMAN; BALDRIDGE, 2007) is a variation which provides a completely transparent interface between surface syntax and underlying semantics. Each (complete or partial) syntactic derivation corresponds directly to an interpretable structure. This allows CCG to provide an account for the incremental nature of human language processing.

The main attraction of using CCG for parsing is that it facilitates the recovery of the non-local dependencies involved in constructions such as extraction, coordination, control, and raising (STEEDMAN; BALDRIDGE, 2007) (JULIA; MARK, 2007).

CCG is a linguistically expressive, but efficiently parseable, lexicalized grammar formalism that provides a “surface-compositional” syntax-semantics interface, in which monotonic rules of semantic composition are paired one-to-one with rules of syntactic composition. The corresponding predicate-argument structure or logical form can therefore be directly obtained from any derivation if the semantic interpretation of each lexical entry is known (JULIA; MARK, 2007).

To illustrate that, a parse of “she bought and sold shares” is shown in Figure 4.1. It presents the application of CCG rules and their range, for instance, the verb “bought” requires an NP (after and before) and the pronoun “she” is a NP.

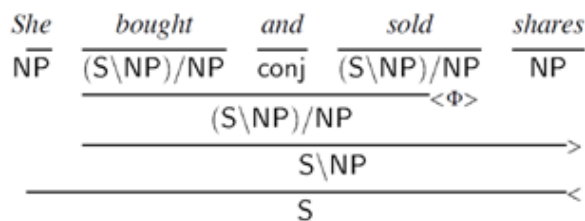


Figura 4.1 – CCG derivation history

In categorial grammar, words are associated with specific categories which define their syntactic behavior. A set of universal rules defines how words and other constituents can be combined according to their categories. In general, the set of syntactic categories is defined recursively as follows:

- Atomic categories: the grammar for each language is assumed to define a finite set of atomic categories, usually S, NP, PP, and VP.
- Complex categories: if X and Y are categories, then X/Y and $X \backslash Y$ are also.

Complex categories X/Y or $X \backslash Y$ are functors with an argument Y and a result X. Here we use a directional categorial grammar, which differentiates between arguments to the right of the functor (indicated by the forward slash “/”) and arguments to the left of the functor (indicated by the backslash “\”). In directional categorial grammar, there are two versions of function application, respecting the directionality of the slash in the syntactic category. However, their effect on the semantic interpretation is the same: Forward Application ($X/Y : f$ $Y : a \implies X : f(a)$) and Backward Application ($Y : a$ $X \backslash Y : f \implies X : f(a)$). In (STEEDMAN; BALDRIDGE, 2007) Steedman and Baldrige advocate the following principles to

which all combinatory rules must adhere in order to keep the generative power of the grammar under control:

- The Principle of Adjacency: Combinatory rules may only apply to string-adjacent entities.
- The Principle of Consistency: All syntactic combinatory rules must be consistent with the directionality of the principal function.
- The Principle of Inheritance: If the category that results from the application of a combinatory rule is a function category, then the slash defining directionality for a given argument in that category will be the same as the one(s) defining directionality for the corresponding argument(s) in the input function(s).

Composition allows two functor categories to combine to form another functor, whereas type-raising is a unary rule which reverts the roles of functor and argument by allowing an argument category X to change into a functor category $T/(T\backslash X)$ (or $T\backslash T/X$), where $T\backslash X$ can be instantiated by any functor category that takes X as argument.

- Forward Composition: $X/Y : f Y/Z : g \implies X/Z : \lambda x.f(g(x))$
- Forward Crossing Composition: $X/Y : f Y\backslash Z : g \implies X\backslash Z : \lambda x.f(g(x))$
- Backward Composition: $Y\backslash Z : g X\backslash Y : f \implies X\backslash Z : \lambda x.f(g(x))$
- Backward Crossing Composition: $Y/Z : g X\backslash Y : g \implies X/Z : \lambda x.f(g(x))$
- Forward Type-raising: $X : a \implies T/(T\backslash X) : \lambda f.f(a)$
- Backward Type-raising: $X : a \implies T\backslash(T/X) : \lambda f.f(a)$

In this work, the syntactic categories, like noun, verb, preposition, for each of the words in a sentence are obtained using the parser PALAVRAS (BICK, 2000) as a part-of-speech (PoS) tagger in a preprocessing step. We adopted this approach to allow the evaluation of the parsing process independently of the PoS tagging process.

4.3 Corpus resources

There are not many large scale annotated corpora resources for Portuguese. Floresta Sintática (syntactic forest) is a publicly available treebank for Portuguese ¹. This corpus is subdivided in 4 corpora (Bosque, Selva, Amazônia and Floresta Virgem). Bosque ² contains 9,368

¹it was created as a collaboration project between the VISL project, and Linguateca (formerly the Computational Processing of Portuguese project). Available at <http://www.linguateca.pt>

²Bosque in English: grove

sentences from CETENFolha and CETEMPúblico newspapers. Since 2007 Bosque has been under manual revision, which includes corrections to the annotation, but which also introduces noise. Therefore, this corpus was a good candidate for the automatic extraction of a CCG grammar to Portuguese carried out in this work (presented in Section 4.2). Figure 4.2 shows an example of Bosque sentence, “*Veículos de resgate estavam a apenas 500 metros de o Airbus 300*”, and, as a matter of illustration, Figure 4.3 shows its equivalent in a classic tree format.

```

STA:fcl
SUBJ:np
=H:n('veículo' <np-idf> M P)   Veículos
=N<:pp
==H:prp('de')      de
==P<:np
===H:n('resgate' <np-idf> M S) resgate
P:vp
=MV:v-fin('estar' IMPF 3P IND) estavam
SC:pp
=H:prp('a')        a
=P<:np
==>N:adjp
===>A:adv('apenas') apenas
===H:num('500' <card> M P)   500
==H:n('metro' <np-count?> <np-def?> M P)   metros
==N<:pp
===H:prp('de' <sam->)      de
===P<:np
====>N:art('o' <-sam> <artd> M S)   o
====H:prop('Aibus_300' M S)   Aibus_300
    
```

Figura 4.2 – An example of Bosque corpus (original)

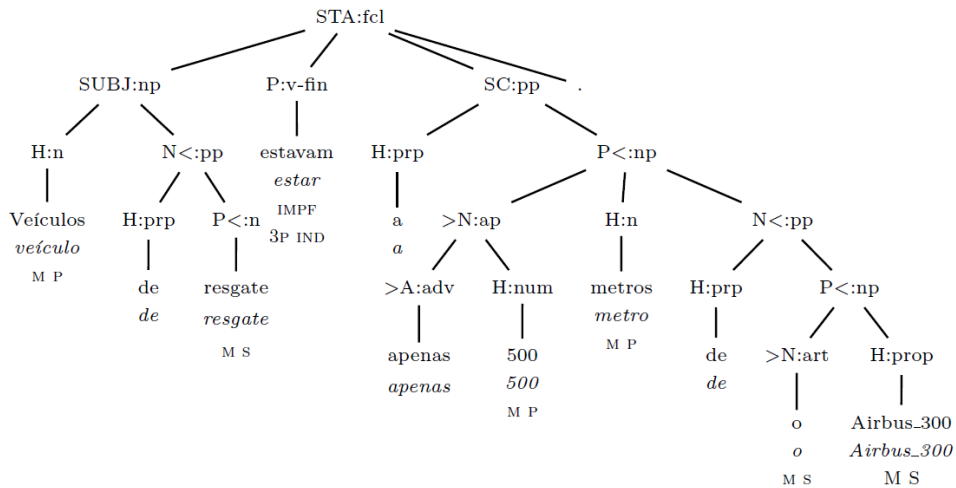


Figura 4.3 – An example of the Bosque corpus in a classic tree format

4.4 Ontology

An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary

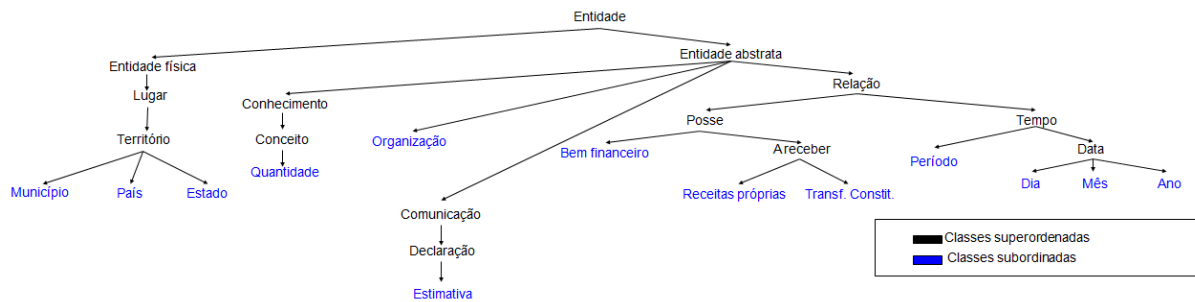


Figura 4.4 – Ontology of FAMURS domain

(ASUNCION; MARIANO; OSCAR, 2004). Ontologies are used by people, databases, and applications that need to share domain information (a domain is merely a specific subject area or area of knowledge, such as petroleum, medicine, tool manufacturing, real estate, automobile repair, financial management, etc.). Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them. They encode knowledge in a domain and also knowledge that spans domains. In this way, they make that knowledge reusable. Since the current work is linked to the automatic answer of constitutional transfers from Federação das Associações de Municípios do Rio Grande do Sul³ (FAMURS) domain, a survey was conducted with domain experts and using information available from the organization to search for data to build an ontology. The survey was done through a personal scheduled visit to the head office of FAMURS, which provided the information needed to select items that should be present in the domain's ontology.

The concepts represented in the ontology cover the main concepts addressed by the sector of constitutional transfers in FAMURS. Examples of concepts are of constitutional transfers, dates and municipalities. The purpose of the construction of this ontology is to enable the identification of concepts in the sentences of the user and to identify which question is being made - and what information is required. For instance the sentence “Qual o FPM de Porto Alegre?” would be analyzed as *constitutionalTransfers(FMP) town(Porto Alegre)*. A simplified version of the ontology is shown in Figure 4.4, which represents the main concepts and their hierarchical relations.

³Federation of Associations of Municipalities of Rio Grande do Sul

5 BASE SYSTEMS

A chatterbot or conversational agent can be equipped with a QA system for dealing with natural language interaction. In this chapter we evaluate the performance of a QA system in terms of the resources employed (shallow vs deep approaches), in section 6.1. We then investigate in particular the contribution of a parser for language understanding. In section 6.2 we discuss the evaluation of the developed system focusing on two aspects: parser performance and dialogue quality. The parser evaluation shows the statistical as well as manual assessment of the learning process to determine the correctness of the system. In the dialogue analysis we show the system evaluation regarding its responsiveness.

5.1 Question Answering for Portuguese: how much is needed?

As there is no consensus on the tools and resources to use for QA, especially for resource-poor languages, we now describe a comparison between shallow and deep tools ¹.

5.1.1 Materials and Methods

The goal of this investigation is to identify the impact of the processing stages, tools and resources employed in the performance of a QA system. In order to do that we developed a standard system with the most common modules used in different systems, and evaluated the performance of the different configurations over a single dataset, the OLinCom corpus.

The OLinCom corpus consists of a set of questions and documents prepared for the First Brazilian Olympiad on Computational Linguistics (OLinCom)². This dataset contains 20 documents and 30 questions, which given its diversity allows the identification of classes of questions that are more difficult to answer for the different configurations, despite its small size in comparison to the corpora used in campaigns such as TREC and CLEF. The corpus contains 4936 tokens distributed in 267 sentences.

The standard QA architecture defined is shown in figure 5.2, and it has three modules: question analysis, document retrieval and passage retrieval. The system receives as input a question (used by the question analysis and document retrieval modules) and a corpus (used by the document retrieval module). For comparing the performance of shallow and deep methods

¹This comparison was published as (WILKENS; VILLAVICENCIO, 2010).

²<http://www.nilc.icmc.usp.br/arianidf/olincom/trilha1.html>

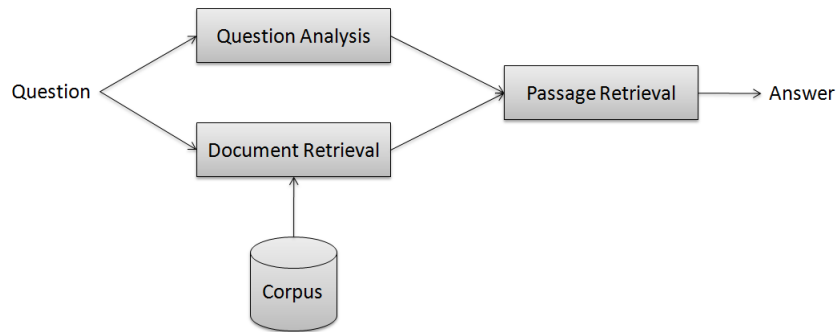


Figura 5.1 – System architecture developed.

for QA the system has several versions of each module, each version using a different technique.

For this module we look at the benefits of the adding deeper information to the rules. In order to do that we compare the use of two sets of rules:

- Shallow rules are based on surface information (words), whereby a mapping is defined between an interrogative pronoun and the expected answer type.
- Deep rules are constructed from a deeper analysis of the corpus, using a decision tree classifier (with the J48 algorithm (QUINLAN, 1993)) for determining the expected answer type. The decision tree was trained on a set of 384 questions (from CLEF 2008) manually annotated with the expected answer type on the basis of both word forms and their grammatical classes.

The document retrieval module can be divided into two stages: generating the query and identifying relevant documents. For the first stage (a) all stopwords are removed, (b) terms are normalized to be used in a canonical form, (c) term expansion is performed and finally (d) the query is formatted. These steps are executed for helping the information retrieval (IR) tool to identify the most relevant documents for the query. For this work, information retrieval is performed using Lucene³, a standard IR tool, in its basic configuration of a vector space model and a stemmer for Brazilian Portuguese. The documents retrieved by the tool are ranked in order of relevance and passed to the passage retrieval module. In order to avoid a high processing costs in the next module, only the most relevant document is considered.

In order to maximize the data available, the corpus was pre-processed to reduce the number of variants and synonyms of each term, and these were annotated with morphosyntactic information (using the parser PALAVRAS).

The passage retrieval module takes the document retrieved by the IR module and splits it into sentences. Based on the expected type of answer determined by the question analysis module, these sentences are filtered to remove unrelated sentences.

³<http://lucene.apache.org/java/docs/index.html>

Question Type	Shallow Rules	Deep Rules
Quote	100%	83%
Place	33%	33%
Entity	7%	46%
Numeric	33%	50%

Tabela 5.1 – Percentage of correct answers per type of questions

The expected type of answer also determines the appropriate procedure that the module follows. If the required answer is a factoid, a named entity recognizer and a filter for numeric or date information are used. On the other hand, if the expected answer is not a factoid, the module uses (1) ngram comparison, (2) POS comparison of the words in the question and those in the candidate sentence and (3) comparison of subject, predicate and verb of the question and candidate answer (referred to as NP-PP-VP in the text). The candidate passages (both factoid and non-factoids) are ranked using bag of words.

5.1.2 Results

To identify the impact of each technique, we tested each with perfect input, by manually cleaning the output of the previous module. After we identified the best methods for each module, we applied the test system as a whole in order to identify the overall performance of the system.

The first module, which processes the question, was tested with the whole corpus. For these 30 questions it obtained an overall accuracy of 36% with shallow rules and 53% with the deep rules (patterns involving e.g. POS tags). Table 5.1 shows the results considering the types of questions, in table are shown a best result using deep rules, in other words, rules machine learning-based. The main advantage of using a parser is the flexibility it allows of combining different levels of information in the patterns, like lemmas and POS tags. One example is the following rule that matches sentences where the pronoun “quem” followed by a verb (*pronoun*=“quem” & *verb*) like *Quem conseguiu uma virada sobre Ksenia Pervak?* and *Quem é o técnico do Corinthians?*. These examples from the corpus cannot be not captured properly by the simple pattern. The document retrieval module was tested according to the amount of information provided about the input question:

- Shallow question uses all the words in the question directly for performing IR
- Deep question extracts nouns terms from the parsed question

Method	Shallow Question	Deep Question
Lucene	90%	83%
Lucene-BR	90%	90%
Pre-processed corpus and Lucene	70%	86%
Pre-processed corpus and Lucene-BR	86%	86%

Tabela 5.2 – Performance of the document retrieval module

Question Type	Approach	Results
factoids	entity recognizer	73%
non-factoids	bag of words	63%
non-factoids	ngrams	45%
non-factoids	POS tag	18%
non-factoids	NP-VP-PP	9%

Tabela 5.3 – Results of factoid and non-factoids question and their respective methods

Table 5.2 show the results obtained. In both cases we evaluated the use of Lucene in its standard version (Lucene) and in the Portuguese version (Lucene-BR). In addition we tested them using both the original and the pre-processed corpus. The best results were obtained with the (complete) shallow question, suggesting that using all the words in the sentence increases the chances of finding the answer, and further information may not imply in improvement of results. In addition, using Lucene customized for Brazilian Portuguese improved the results when the deep question was used. The last module, passage retrieval, was tested using as input the correct document for the question, as well as the correct identification of the type of expected answer. The results per type of answer are show in table 5.3.

Finally, we tested the overall performance of the system using two configurations. The first uses the configuration that provided best results at each stage (deep rules, shallow questions with simple Lucene, entity recognizer in factoids questions and bag of words in non-factoid questions), while the second uses the shallow version of each stage. In both cases the overall performance of the system is similar and not significantly different, reaching 56% accuracy.

In sum, testing different configurations, in the absence of more sophisticated tools, a shallow approach presented good results. The results obtained analyzing each phase separately suggest that for the stage of question analysis deep methods perform best, using learning algorithms based both in surface (words) and deep forms (parsing information). However, the use of a pos-tagger, which is more widely available for Brazilian Portuguese could supply the information needed. For document retrieval the best result is achieved with shallow information, using the whole question either with Lucene or with the customized Portuguese version. The latter improves performance even when deep information is used, perhaps compensating for

the size of the corpus. In passage retrieval the best results was with the shallow bag-of-words approach.

5.2 COMUNICA - A Question Answering System for Brazilian Portuguese

This section discusses a question answering system for Brazilian Portuguese, it is relevant in this dissertation because this work is a possible extension of COMUNICA. COMUNICA is a voice QA system for Brazilian Portuguese with search capabilities for consulting both structured and unstructured datasets. One of the goals of COMUNICA work is to help address digital inclusion by providing an alternative way to accessing written information, which users can employ regardless of available computational resources or computational literacy.

The Comunica system is composed of five modules: a manager module and four processing modules, as shown in figure 5.2. The manager is responsible for the integration and communication with the speech recognition, text processing, database access, and speech synthesis modules.

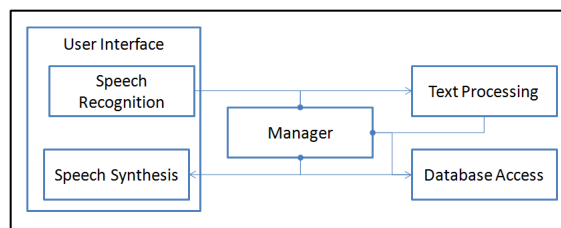


Figura 5.2 – Architecture of the system.

5.2.1 Speech Recognition

For continuous speech recognition of the users' requests the system uses an automated phone service. This module uses two research fronts signal analysis (Fourier transform and Wavelets). The coefficients obtained are sequenced on three fronts for continuous speech recognition: HMMs (BECERIKLI; OYSAL, 2007) TDDNN and NESTOR (NASUTO; BISHOP; DEMEYERC, 2009). To train the models, a corpus of FAMURS callcentre telephone interactions has been recorded. The recognition focuses on the vocabulary employed in the domain, in this case municipal information related to taxes from FAMURS. In order to do that, it uses 2 ontologies to validate the candidate words in the input: (a) a general purpose and (b) a domain ontology. The recognised transcribed input is passed to the manager for further processing.

5.2.2 Text Processing

The manager sends the transcribed input to be processed by the natural language processing module. The natural language queries are processed using shallow and deep tools and accessing both a general and a domain specific ontologies (illustrated in Figure 5.3). This module needs to determine which type of query the user performed and what is the likely type of answer, based on mostly lexical and syntactic information. This process is divided into 3 main steps: parsing, concept identification and pattern selection. In the first step, the input is parsed using the PALAVRAS parser (BICK, 2000), and the output provides information about the particular pronoun (wh-word), subject and other verbal complements in the sentence. For concept identification, the system uses the domain ontology, which contains the relevant concepts to be used in next steps. The ontologies also provide additional information about nouns (such as hyperonymy and synonymy) for determining which instances of the concepts were present in the input. For example, “Gramado” is an instance of the concept “city”. Both absolute and relative dates and periods (e.g. last quarter, first week) need to be treated.

Finally, based on this information this module selects from a set of pre-defined question patterns linking concepts of the domain ontology with SQL commands, the one which contains the largest number of concepts in common with the input, and sends it to the manager in an XML format. If there is no complete frame, this module identifies which concepts are missing and returns this in the XML output.

5.2.3 Database Access

The search module is divided in two sub-modules: one for searching information in a structured database and other for searching in an unstructured knowledge base. It receives as entry an XML file, containing the original input in natural language and the concepts identified in the question. The structured search module receives the input tagged with concepts of the ontology and an identified search pattern, and selects a structured SQL query. These queries are predefined according to the search patterns and the structure of the database. For example, in the case of the FAMURS domain, there are concepts related to time period, cities and taxes. When these 3 concepts are found in the input, a special pattern is selected which defines the kind of information that must be retrieved from the database. An SQL command is then executed in the structured database. All possible patterns are mapped to a specific SQL command. These commands have slots that are filled with instances of the concepts identified in the sentence.

For example, names of cities are instances of the concept “city”. The retrieved values are used for producing the answer in natural language, using some predefined answer patterns.

Otherwise, the system uses the ADS Digital Company Virtual Assistant (VA) (DUIZITH et al., 2004) to search the unstructured data (e.g. Frequently Asked Questions), using the lexical information to locate the answer associated to the most similar question. This answer is written in natural language and will be returned to the main module of the system. If no similar question is found according to a predefined degree of similarity, the VA returns a standard answer.

5.2.4 Speech Synthesis

The text output to the user is synthesized, resulting in an audio file that is transmitted through the server.

5.2.5 Manager

The manager is responsible for the integration and communication of the modules. It processes requests, interpreting the actions to be taken and dispatching the requests to specific modules. To start the interaction the manager activates the speech recognizer, and if no problem is detected with the input, it is passed to the text processing module. In the case of missing information, the manager informs the user that more information is needed. Otherwise, the query is passed to the database module. The database module then returns the result of the query to the manager, which sends this information to the interface component.

All the components are designed as Web services. This allows us to use a common and simple way of communication among components, allowing a certain degree of independence. Hence components can be implemented using different technologies and may be distributed among different servers, if needed.

5.3 Grammar Engineering

Hockenmaier and Steedman (JULIA; MARK, 2007) presented a method for obtaining a corpus of CCG derivations and dependency structures from the Penn Treebank ⁴. Earlier

⁴A treebank or parsed corpus is a text corpus in which each sentence has been parsed, i.e. annotated with syntactic structure.

versions of the resulting corpus, CCGbank, have been used to build a number of wide-coverage statistical parsers, which recover both local and long-range dependencies directly and in a single pass.

Moot (MOOT, 2010) describes the development of a wide-coverage type-logical grammar for French, which has been extracted from the Paris 7 treebank and received a significant amount of manual verification and cleanup. Tse and Curran (TSE; CURRAN, 2010) present a Chinese CCGbank, a 760,000 word corpus annotated with Combinatory Categorical Grammar (CCG) derivations, induced automatically from the Penn Chinese Treebank (PCTB).

5.3.1 Algorithm

Floresta Sintática (syntactic forest) is a good candidate for automatic extraction of a CCG grammar to Portuguese. Figure 5.4 shows an example of Bosque headline sentence, “PT⁵ em o governo” (PT in government). In this example a syntactic tree is displayed in flat form in which PT (a proper noun) is the head of a NP (noun phrase), em (a preposition) is the head of a PP (a prepositional phrase) and this PP has a nested NP (composed of an article and a noun). In this figure each constituent is displayed in a line, with its syntactic role in capital letters, its full morphosyntactic annotation after the “:”, the constituency information marked by “=”, and the surface form of the word. For example, =*H:prop('PT' M S)* PT represents the proper noun (*prop*) “PT”, with canonical form “PT”, that is a masculine (*M*) and singular (*S*).

The corpus was preprocessed and reformatted for automatically deriving the grammar. This reformatting was to correct the tree:

- The first correction is the indentation of the root in relation of the rest of the tree which is moved one level to the right. This is exemplified in Figure 5.5b.
- The second correction is with respect to the number of terminal nodes. Following (JULIA; MARK, 2007) we reformatted the corpus in way to take a binary tree. Figure 5.5 which use the lowest number of terminal nodes. For instance, Figure 5.5 shows an input (a) and output (c and d) tree fragment illustrating the process. The tree is processed in order to obtain in each level either a terminal node with other non-terminal nodes or only two terminal nodes or, yet, only non-terminal node⁶. This step is for simplifying possible categories in the grammar. In example the node “NP” which had 3 nodes was modified

⁵PT means “Partido dos Trabalhadores” (in English: Workers Party)

⁶In this work a terminal node in a derivation tree is a part-of-speech and the other nodes are variables (or phrases). In example of Figure 5.5 the terminal nodes are art, n and adj and the variables are fcl, np, adjp.

to 2 nodes.

- Like Hockenmaier and Steedman (JULIA; MARK, 2007), we reformatted the corpus, but in our work we make it manually, because there are many different cases of reformatting. For instance [art n adjp] (tree fragment of Figure 5.5a) has to be reformatted to [np [art n] adjp] and [adj n vpcp] to [np [adj n] vpcp].

The processed corpus algorithm is used for the grammar induction employing the following heuristics:

Direct allocation: We create a list with all leaf nodes and for each parent node we set a CCG rule. If the node has only two terminals as children the rule attribution is the PoS from the parent node followed by a slash and following by the PoS from the second child node. The slash is based on the position of the words in the original sentence, if the first PoS is the parent of the second then the slash is a forward slash, or else it is a backward slash. Figure 5.6 illustrates this step with the leaf node (“art” and “n”), their parent (“np”) and the resulting category *np / n*.

Indirect allocation: This step aims to treat trees which have depth of more than one level (i.e. have subtrees as children). In this step we create a dictionary of nodes which are removed from the tree. Figure 5.7 illustrates an example of indirect allocation, the subtree “np” that already has a CCG category defined, is replaced in the tree by the node “np” then the node “pp” can be analyzed, and by direct allocation the result is “pp / np”.

Removing Children: In this step we prune the tree by elimination of nodes whose parents have a CCG rule defined. The entire pruned node has their CCG rule saved in the rules output file. The example in Figure 5.7, after this step (contains only the “pp” node (the other nodes were removed)).

Semantic allocation: The fourth step, assigning semantics, was performed manually as a later step. In semantic attribution we apply three templates of semantic (modifier, complement and list). The modifier semantic is an attribution like a property of another concept. The complement semantic is an attribution like a property, but applicable only to verbs. The list semantic is a list of attributions linking a set of concepts. However, this allocation can be performed automatically through the use of one assignment of semantics to each of the morphosyntactic categories.

5.3.2 Parser Evaluation

To evaluate the developed parser we performed two types of analysis (statistical and manual). The statistical analysis focused on the similarity between the response of the parser against the gold standard, as well as the quality of learning of categories. The manual analysis considered whether the parser’s output presented a consistent result in terms of understanding.

5.3.2.1 Statistical Evaluation

The standard techniques for evaluating parsers and grammars are called PARSEVAL measures, and were proposed by Becerikli and Oysal (BLACK et al., 1991) based on the same ideas of signal-detection theory. The intuition of the PARSEVAL metric is to measure how much the constituents in the hypothesis parse tree look like the constituents in a hand-labeled gold reference parse. PARSEVAL thus assumes we have a human-labeled “gold standard” parse tree for each sentence in the test set; we generally draw these gold standard parses from a treebank like the Penn Treebank (JURAFSJY; MARTIN, 2009). Given these gold standard reference parses for a test set, a given constituent in a hypothesis parse C_h of a sentence s is labeled “correct” if there is a constituent in the reference parse C_r with the same starting point, ending point, and non-terminal symbol.

$$\text{labeled recall} = \frac{\# \text{ of correct constituents in hypothesis parse of } S}{\# \text{ of correct constituents in reference parse of } S}$$

$$\text{labeled precision} = \frac{\# \text{ of correct constituents in hypothesis parse of } S}{\# \text{ of total constituents in hypothesis parse of } S}$$

$\# \text{ of correct constituents in hypothesis parse of } S$ corresponds to the number of correct constituents labeled, $\# \text{ of correct constituents in reference parse of } S$ is the reference constituents in gold standard and $\# \text{ of total constituents in hypothesis parse of } S$ corresponds to the total number of constituents labeled.

As of the time of this writing, the performance of modern parsers trained and tested on the Wall Street Journal treebank for English is somewhat higher than 90% recall and 90% precision. Instead of measuring how many sentences are parsed correctly, we measured constituent accuracy since it gives us a more fine-grained metric. This is especially true for long sentences, where most parsers do not get a perfect parse. If we just measured sentence accuracy, we would not be able to distinguish between a parse that got most of the constituents wrong, and one that just got one constituent wrong.

In this work we used the Bosque and Amazônia corpora as a gold standard for parser. Since the grammar acquisition was done from Bosque we used Amazônia as gold standard to

investigate the grammar generalization. Because the corpora are in a different format we carried out a manual comparison of categories (from CCG) and the derivation tree (from corpus). The evaluation used these two gold standards to enable a broad analysis of the grammar induction. These gold standards are composed of 100 sentences each one chose randomly from Bosque (1545 words) and Amazônia (2267 words). The statistical evaluation results are shown in Table 5.4 ⁷, where the biggest problem was the rejection of sentences (since many sentences would be accepted if all the extracted grammar would have been used). We also highlight that many sentences (25% in the Bosque and 32% in the Amazônia) were not analyzed for lack of memory during parsing.

Metric	Bosque	Amazônia
# sentences	100	100
# words	1545	2267
Sentences rejected	45%	66%
Sentences successfully parsed	75%	68%
Labeled Recall	51%	56%
Labeled Precision	73%	76%

Tabela 5.4 – Statistical evaluation of the parser

Because of the use of a machine learning approach, it is important to evaluate the rules obtained in terms of the learning curve given the size of the data. In this sense we evaluated the number of categories obtained with the use of corpora with different sizes. To perform this test we fragmented a whole corpus into 9 parts (which present linear growth in size). Thus we see a proportional gain in the number of identified categories shown in Figure 5.8, where frequency is the usage of the role of the rule in the derivations, the rank order is the position of the rules sorted by frequency, and F is a corpus fragment, it being 1 the smallest and 9 the biggest, thus indicating that the use of a large corpus favors a statistical analysis (since it increases the difference between the possible useful categories and random noise).

To assess the gain in each part of speech shown (Figure 5.9) we present frequencies of the rules obtained, showing that the corpus is of a sufficient size to learn combination of rules. Figure 5.9 also shows that learning has a similar Zipfian distribution for all grammatical classes, despite the difference in their frequencies in the corpus (this was expected, since the least used forms have smaller variance). Both figures (5.8 and 5.9) present the same behavior that the learning curves, found by Hockenmaier and Steedman (JULIA; MARK, 2007) in spite of the small difference between the algorithm presented in this work and the one proposed by them

⁷In Table 5.4 the line “Sentences successfully parsed” is the number of sentences analyzed over the number of sentences accepted.

PoS	Number	Example of Category
adj	8	n \ n
adv	15	np / np
art	20	np / n
conjc	16	np \ n / n
conjs	15	np / np
ec	1	n / n
intj	6	np / PONT
MARC	19	np \ np / np
Noun	10	n
num	6	np / n
PONT	87	spont \ np
prondet	9	n / n
pronindp	3	n / np
pronpers	6	n / np
prop	10	np
prp	16	np / np
vfin	28	sfin / sinf
vger	23	sger / np
vinf	21	sinf / np
vpcp	13	spcp / np

Tabela 5.5 – Numbers of PoS resulting of manual evaluation of the categories quality

(JULIA; MARK, 2007) and the outcomes in terms of learning are consistent.

After extracting the categories using the full corpus the rules were manually filtered by possible quality by a human expert. With this step done the number of high quality categories is shown in Table 5.5. However, for the evaluation of the parser we adopted a subset of the evaluated rules (for reasons of reduction in processing time we use at most 10 by rule). In Figure 5.10 we show the curves of frequency of the 10 most frequent categories. The parts of speech used in this work are the annotated in Bosque corpus and that the parser Palavras uses. These parts of speech are: adj (adjective), adv (adverb), art (article), conjc (coordinate conjunction), conjs (subordinate conjunction), ec (prefix), intj (interjection), MARC (markings like commas), noun, num (number), PONT (punctuation), prondet (determiner pronoun), pronindp (independent pronoun), pronpers (personal pronoun), prop (proper noun), prp (preposition), vfin (finite verb), vger (gerund), vinf (infinitive), vpcp (participle).

In this work we opted for grammar induction because there is not a Portuguese standard grammar, or even a set of basic rules. We do not compare our result with other work because we did not find other Portuguese grammar.

5.3.2.2 Human Evaluation

In to order focus on the impact of the parser on the quality of the answer we also report human evaluation. All derivations accepted by the parser were evaluated and only one correct derivation of the sentence to be needed for the sentence was considered successfully parsed. For a more detailed evaluation, looking at different corpora, we divided the process in two stages: Bosque and Amazônia. Bosque represents the training corpus for the parser and Amazônia the held-out test corpus. Each sentence in these corpora is annotated with the POS information for each of the words. We analyze the parseable sentences and 92% had correct derivation in Bosque and 100% in Amazônia. The identification of correctness was performed by human evaluation, and a derivation was considered correct if the derivation tree presented a possible correct relation of words.

5.4 Summary

This section presented the state of the art in question answering systems. For resource-rich languages like English many such systems have been developed involving different levels of processing, such as Javelin (NYBERG et al., 2002), QuALiM (KAISSER, 2005), TrueKnowledge, LAMP (LIN, 2005), Ephyra (SCHLAEFER et al., 2006) and AnswerBus (ZHENG, 2002). QuALiM, for instance, explores the use of lexical resources such as FrameNet, and PropBank VerbNet in QA and considers two different complementary methods that use these resources. Given this context, the next chapter describes resource and tools employed in the proposed work.

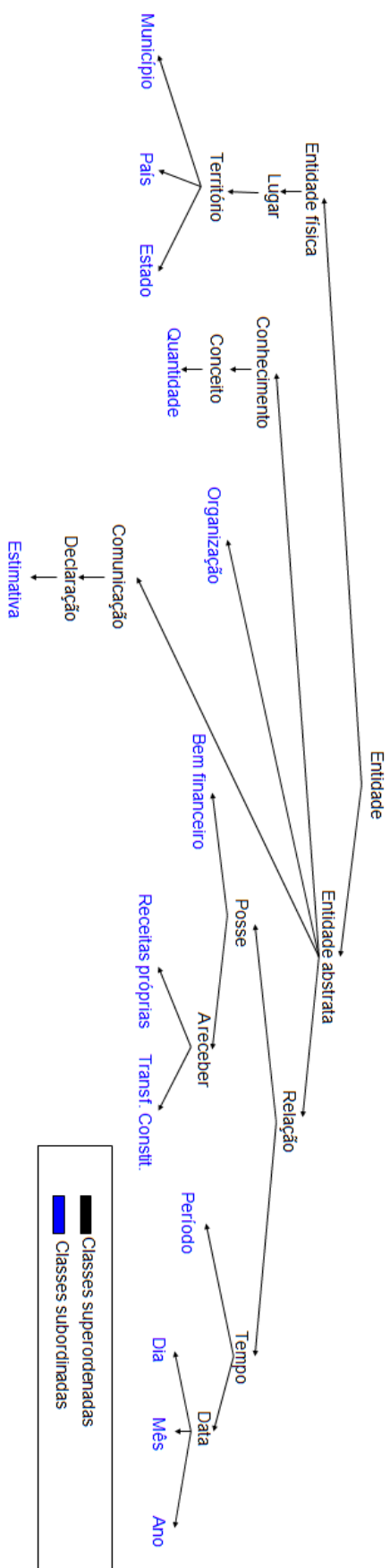


Figura 5.3 – The domain ontology

```

UTT:acl
SUBJ:np
=H:prop('PT' M S) PT
SA:pp
=H:prp('em' <sam->)em
=P<:np
==>N:art('o' <-sam> <artd> M S) o
==H:n('governo' M S) governo
    
```

Figura 5.4 – An example of Bosque corpus

STA:fcl ACC:np =>N:art('um' <arti> M S) um =H:n('dado' <np-idf> M S) dado =N<:adjp ==H:adj('surpreendente' <ALT> M S) supreendente	STA:fcl =ACC:np ==>N:art('um' <arti> M S) um ==H:n('dado' <np-idf> M S) dado ==N<:adjp ===H:adj('surpreendente' <ALT> M S) supreendente
--	---

(a) Original (fragment) tree

(b) Correction of indentation

STA:fcl =ACC:np == AUX:Np ===>N:art('um' <arti> M S) um ===H:n('dado' <np-idf> M S) dado ==N<:adjp ===H:adj('surpreendente' <ALT> M S) supreendente	STA:fcl = np == np ===> art/um === n/dado == adjp === adj/supreendente
---	--

(c) Output (fragment) tree

(d) Output (fragment) tree (Just PoS)

Figura 5.5 – Example of the 4 steps of corpus reformatting

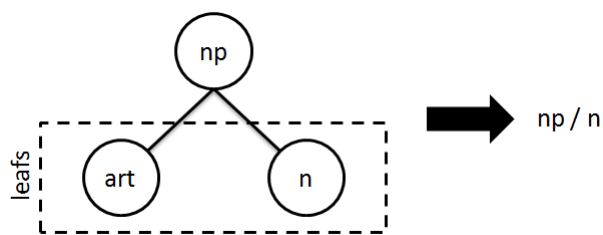


Figura 5.6 – Example of direct allocation step

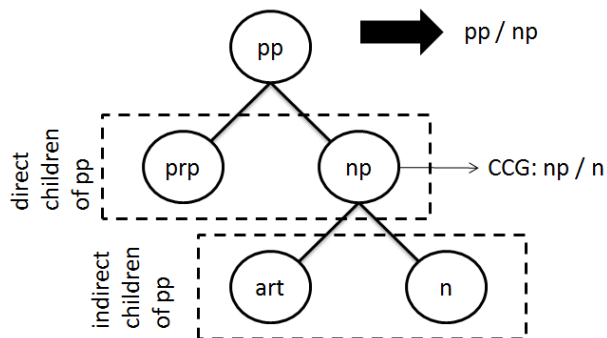


Figura 5.7 – Example of indirect allocation

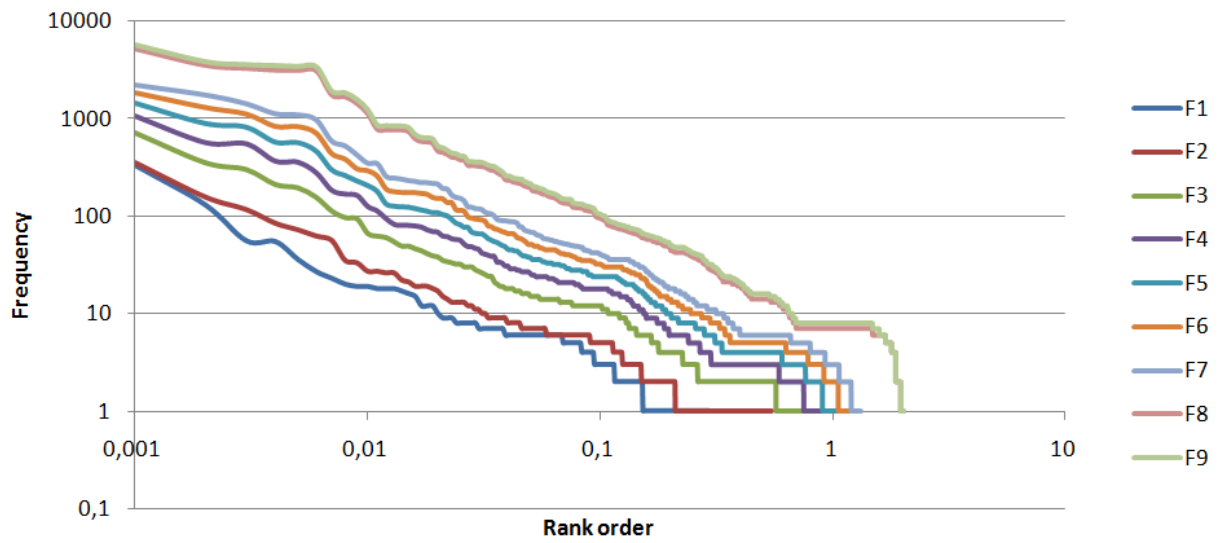


Figura 5.8 – Comparison of learning category combination rules from 9 fragments

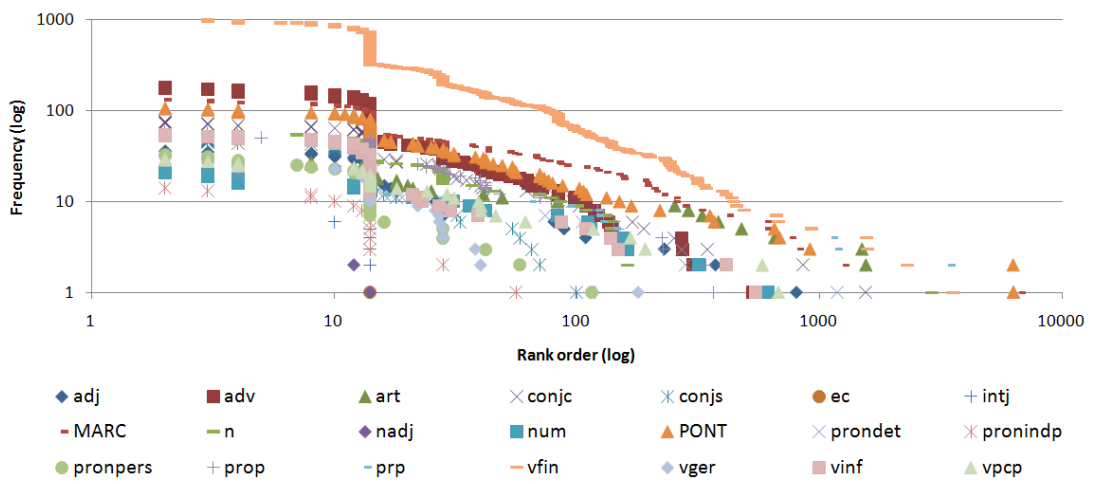


Figura 5.9 – Learning curve of PoS

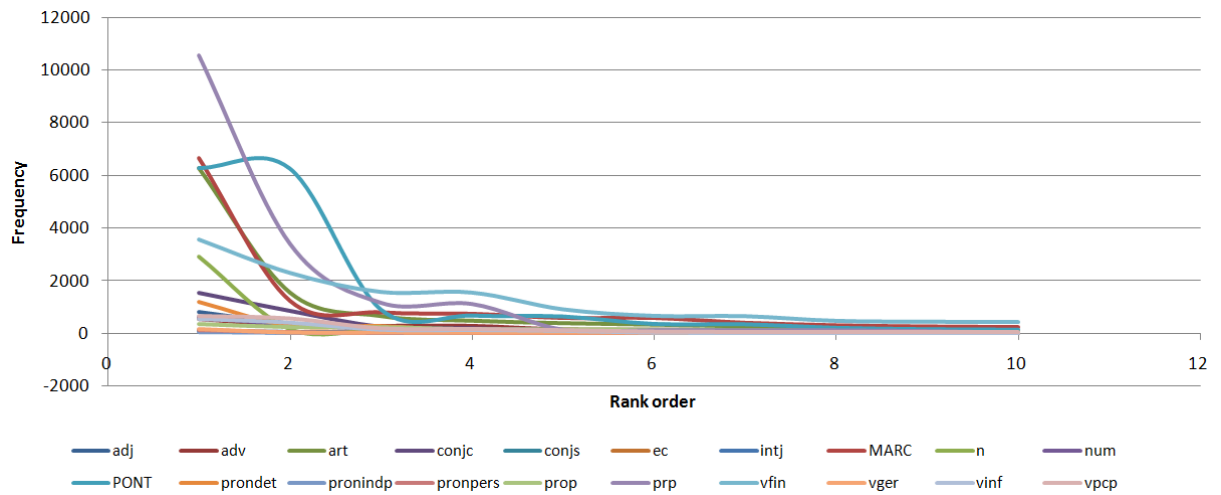


Figura 5.10 – Frequency of class after evaluation manual

6 ARCHITECTURE

To develop a system to evaluate the contribution of parsing systems for conversational agents we opted for a system of real conversation. In this way we use the constitutional transfer domain. A similar work in this domain is the COMUNICA project, that is a voice QA system for Brazilian Portuguese with search capabilities for consulting both structured and unstructured datasets. One of its goals is to help address digital inclusion by providing an alternative way of accessing written information, which users can employ regardless of available computational resources or computational literacy. Because of this, we used the Comunica design, due to its focus on a phone answering service delivered to FAMURS.

This section explains the developed dialogue manager system, and presents the conversational plans and task manager (including the relationships developed between plans and the additional systems).

6.1 Conversational Plans

In order to relate the actions of the agent to the place (database and text file) of the requested information we used BDI plans, which have the choice of activation based on the agent's beliefs. For example, if the agent has a belief that a question was asked and also that it should use constitutional transfer, municipality and date to answer, then the agent will perform a search on the database and return the information.

For the creation of plans we originally used a process of question patterns creation. To do this, we employed the linguistic knowledge of the domain in order to define the maximum number of possible patterns for a later selection. The initial approach was based on the definition of syntactic patterns, containing interrogative pronouns and number of phrases in each question pattern. A total of 132 complex patterns of questions were defined by a linguist and the most frequent were selected based on the majority of issues faced by FAMURS for five patterns of questions with the answers expected (this five patterns are presented in Table 6.1). Example of a pattern is “Quanto <*município*> vai receber de <*transferencia constitucional*> em <*data*>?”.

These patterns were complemented on the basis of analysis of data contained on the transcription of two weeks of calls to FAMURS. In this analysis we considered communicative acts, solicitations and the FAQ database, and each one of these were related to an action triggered by a plan.

A dialogue example is presented in Figure 6.1. The sentence, its translation by the

Pattern Name	Pattern	Example
Value of one transfer	município transferencia constitucional data(dia, mês, ano)	Quanto Bagé vai receber de ICMS no dia vinte e três de julho de dois mil e dez
Value of all transfers	município transferencia constitucional data(mês, ano) quanto	Quanto Bagé vai receber no dia quinze de janeiro de dois mil e dez
Next Date	município transferencia constitucional	Quando Bagé vai receber a terceira parcela do FPM de julho de dois mil e dez?
All dates	município transferencia constitucional data(mês, ano) quando	quando Bagé vai receber as parcelas do FPM de julho de dois mil e dez?
Answer is a date	município transferencia constitucional data(dia, mês, ano) 1parcela	Quando Bagé vai receber a 1 parcela do ICMS do mês de abril de 2010?

Tabela 6.1 – Five Patterns

parser, the module responsible for answering and the given answer are shown. The dialogue exemplified in 6.1 illustrates the main information flow in this architecture: the sentence (user input); parsed input (user input in logical form); module call (module identified by BDI to be active); and answer (agents answer). The *X* in the logical form indicates the identifier of the entity and *object* corresponds to the identification of a concept.

Sentence	Parser	Answer
Oi	object(oi) ^ object(.)	Oi! Qual o tipo de informação você deseja?
Por favor.	object(por,favor) ^ object(.)	Sim?
O que é ICMS?	é(X_14) ^ ICMS(X_11) ^ object(?) ^ object(o,que)	Empty
Qual o ICMS de Canoas amanhã?	é(X_14) ^ object(qual) ^ object(ICMS) ^ object(?) ^ object(amanhã)	R\$10000
Obrigado	object(obrigado) ^ object(.)	De nada
Tchau	object(tchau) ^ object(.)	Tchau

Figura 6.1 – Example of dialogue and system internal messages

6.2 The Proposed Alice

For the realization of communication unrelated to the conversation domain we developed a chatterbot ALICE. We chose to use chatterbots for two reasons:

- They favor the representation of adjacent turns;
- We consider it important to show the relationship of architecture and chatterbots in this work, due to the existence of conversational agents that are focused on nonverbal communication and with a chatterbot on the conversational module.

Unlike the original version (in English), where the AIML scripts have thousands of patterns, and which is capable of holding a brief conversation, this script has incorporated only forms of saying hello, goodbye and thanks. We chose this simplified version because of the scope of our work, since the development of these scripts are an expensive task in a matter of time and complexity and the focus here is on the analysis of parsing resources needed for conversational agents.

Despite the simplifications, the interaction between chatterbot and BDI simplify the modeling of dialogue, for it furthers the development of plans (for segments of dialogue directly related to the field) and the development of adjacent turns (for the segments of dialogue related to the dialogue structure).

6.3 The Proposed Question Answering System

Since part of the necessary information for the conversational agent is available in text file as unstructured information a system of QA (for Portuguese) was developed. This system was based on the analysis of Wilkens and Villavicencio (WILKENS; VILLAVICENCIO, 2010) where the resources necessary to develop a QA system for languages with limited linguistic resources are compared.

Our question answering system has three modules (question analysis, document retrieval and passage retrieval). In the question analysis we used a simple system that associates the interrogative pronoun with a determined type of question. In the document retrieval module we used the LUCENE¹ system for information retrieval. In passage retrieval we compared the bag-of-words (weighted frequency of terms in common) between the question and the candidate answer.

The set of documents that serve as the basis of the QA system contain information for questions frequently asked to the FAMURS's personnel. We chose to use a QA system for these documents, because although there is some association between questions and answers, these answers are ambiguous and could be used for several different questions. Thus the use of QA

¹ Available at <http://lucene.apache.org>

allows flexibility regarding the entry of questions on the system.

6.4 Task Manager

The internal actions of the agent are controlled in the task manager module, which is integrated into the dialogue manager module. When the agent has some belief a BDI plan is triggered, directing the agent to accomplish a goal (perform a task). The three actions to be performed by the conversational agent are accessing a database, retrieving information and performing general communication. Access to database is through a simple connection to the database, since the NLU process identifies the elements necessary to the query (tables, fields and filters). The information retrieval uses the QA module (searching the user's question in the set of documents). The rest of the communication is conducted by ALICE, which contains coded pairs of system dialogues (important information for conversation, but not for the specific task of the agent).

6.5 Summary

To analyze the contribution of parsing for conversational agents we have developed a pipeline of modules covering the NLU and dialogue manager modules (task manager module is integrated into the dialogue manager). This required the development of a CCG grammar for Portuguese, as well as a specific ontology for the dialogue domain. For inducing the Portuguese CCG grammar, we used a translation from tree Bosque form to CCG categories, and then we used the category frequencies to identify the most probable categories. As we selected the domain, we opted for the constitutional transfers of FAMURS, because of its transcription of real interaction with people actually asking for information. Another important aspect of this domain is that it covers the entire State of Rio Grande do Sul, thus having also information on regionalism.

The NLU module has two major steps (parser and concept identification). In the parser step the system identify the PoS of each word using the Palavras parser, then the Palavras output is sent to CCG parser where the logical form is identified. In the concept identification we used the domain ontology to identify what word are concepts in ontology (e.g. the word "Porto Alegre" is a "city", a "place", a "physical entity" and an "entity").

The dialogue manager module uses the NLU output to identify what action needs to be

done (e.g. access data-base, query to QA system). Once the action is identified the system calls the subsystem responsible to perform.

A dialogue example was presented in Figure 6.1. The sentence, its translation by the parser, the module responsible for answering and the given answer we are shown. The dialogue exemplified in 6.1 illustrated the main information flow in this architecture: the phrase (user input); parsed input (user input in logical form); modulo call (module identified by BDI to be active); and answer (agent's answer).

7 EVALUATION

This work focuses on the contribution of the parser as part of developing conversational agents for the NLU process. In this chapter we discuss the evaluation of the developed system. This evaluation focused on two aspects: parser performance and dialogue quality. The parser evaluation shows the statistical as well as human assessment of the learning process to determine the correctness of the system. In the dialogue analysis we show the system evaluation regarding its responsiveness.

7.1 Dialogue

The dialogue evaluation considered the correctness of the answer given to the user. To do this we analyzed the NLU process and their output to dialogue manager.

Following Isbister and Doyle (ISBISTER; DOYLE, 2011) we adopt their evaluation criteria for conversational agents in terms of categories of research, success criteria and evaluation techniques (Table 7.1). In this work we focused the evaluation on a domain-specific task for a ‘real-world’ application.

In the field of human-computer interaction, a Wizard of Oz experiment (KELLEY, 1985), is a research experiment in which subjects interact with a computer system that subjects believe to be autonomous, but which is actually being operated or partially operated by an unseen human being.

The term Wizard of Oz (originally OZ Paradigm) has come into common usage in the fields of experimental psychology, human factors, ergonomics, linguistics, and usability engineering to describe a testing or interactive design methodology wherein an experimenter (the “wizard”), in a laboratory setting, simulates the behavior of a theoretically intelligent computer application (often by going into another room and intercepting all communications between participant and system). Sometimes this is done with the participant’s a priori knowledge and sometimes it is a low-level deceit employed to manage the participants expectations and encourage natural behaviors.

For example, a test participant may think he or she is communicating with a computer using a speech interface, when the participants words are actually being secretly entered into the computer by a person in another room (the “wizard”) and processed as a text stream, rather than as an audio stream. The missing system functionality that the wizard provides may be implemented in later versions of the system (or may even be speculative capabilities that current-day

Category	Criteria for Successes	Evaluation Techniques
Believability	Agent conveys 'illusion of life' to the viewer/user.	Subjective: Does the user find the agent's appearance, voice, and words, and reactions and answer believable? Does an expert? Objective: Does the user react physiologically and behaviorally as if dealing with an equivalent 'real' person? Does the user engage in ways that demonstrate s/he treats the agent's behavior as believable (reactions to behaviours, attribution of goals and emotions).
Sociability	User is able to interact socially in an intuitive and natural way with the agent.	Subjective: Qualitative measures form user of agent's friendliness, helpfulness, social qualities, communication abilities. Also, user's evaluation of overall experience - speed, ease, satisfaction. Objective: Measures of elicited social responses to the agent. Behavioral changes predicted by social tactics used (more influence of agent on user's answers, more reciprocal aid of agent, etc).
Application domains	Agent performs domain-specific role in a manner that achieves the desired outcome and creates a satisfying, experience for the participant.	Subjective: Measures of user satisfaction with task and interaction. Objective: Behavioral outcomes (performance on tasks, memory, etc.).
Agency and computational issues	System/technique meets good design criteria and performance benchmarks. Also: believability and sociability goals above.	Subjective: Elegance of system, parsimony. Objective: Successful operation of the agent in 'real-world' domains according to criteria of speed, efficiency, optimality, reliability, error handling, etc.

Tabela 7.1 – Major categories of Embodied Conversational Agent research

Fonte: (ISBISTER; DOYLE, 2011)

Evaluation	NLU	Dialogue Manager
Transcriptions with parser	88%	70%
Transcriptions without parser	88%	70%
Comunica with parser	79%	65%
Comunica without parser	89%	68%

Tabela 7.2 – Explicit Dialogue Evaluation

systems do not have). In testing situations, the goal of such experiments may be to observe use and effectiveness of a proposed user interface by the test participants, rather than to measure the quality of an entire system. This experiment approach is used in works like (ROJC et al., 2007) and (BROWN; BARRETT, 2006).

Inspired by the Wizard of Oz approach, we used the transcription of real people dialogues requesting information from FAMURS. These dialogues were recorded over a period of two weeks and transcribed by a linguist in the context of the project Comunica. Of the 25 transcripts we had to dispose of 15 for not addressing the scope of this work, thus resulting 10 transcripts (370 sentences in total).

We used the transcripts as a parameter to assess the internal work of the developed agent by evaluating the agent’s capacity of responding correctly. We analyzed the comprehension of language (the correct operation of the parser), the dialog management (identification of the correct action) and the agent answer (correctness of answer of the enabled module).

Aiming explicitly to evaluate of the conversational agent in the FAMURS domain we used the set of question patterns described in Section 6. As for the evaluation of the dialogue, we evaluated the ability of the agent to understand the sentence using a parser, achieving the correct action and appropriate answer. The evaluation results are presented in Table 7.2. It presents the performance in a test using only the Comunica Project test cases and the performance using transcriptions (with and without the parser).

Analyzing the results it is observed that the use of the parser yields a result similar to the result without it. However, using a parsing system allows the use of BDI plans to do a more specifiable task. One example is the pattern “Quando <munícipio> vai receber a primeira parcela do <transferencia constitucional> em <data>” that, when compared to the pattern “Quanto <munícipio> vai receber de <transferencia constitucional> em <data>?”, identifies that the use of the ontology alone would present a lack of differentiation between the patterns. In a qualitative analysis of our results we identify that the wrong answers were because of the indirect questions (i.e. “Vocês já tem a previsão pro FPM do dia vinte e trinta?”) and the failure

in parsing step. Possible improvements are to add more rules in grammar used by the parser and complement the agent's plans.

7.2 Summary

In this chapter we presented an evaluation performed on the developed system. This evaluation focused on two points: dialogue analysis and parser analysis. In the analysis of the dialogue we concentrated on the capacity and quality of the response of the agent, while in the parser analysis we focused on performance.

The developed system is composed of (1) the NLU component (the CCG parser to translate from NL to an equivalent logic form); (2) an ontology used to identify the concepts in user sentence; and (3) a dialogue manager (the BDI plan identifies the action to be carried out and the possible actions: QA, data-base and Alice response).

To evaluate the parser we tested it with two sets of texts (from Bosque and Amazônia), each one with 100 sentences. The result of the analysis proved not to be satisfactory, but the manual evaluation showed that the derivations obtained are not exactly the same as those presented in the training corpus, but they are correct derivations. In this sense it is important to use a system that enables the use of a greater number of categories than ours.

The evaluation of dialogue measured the response quality, by analyzing the pipeline performed internally by the agent, as well as the compliance of the response to the input sentence. In order to simulate a more realistic evaluation we used recordings of actual dialogues held at FAMURS, which were transcribed and used as input to the agent, and the agent's responses were considered correct. The agent with and without the parser shows a similar performance in the quality of answers (both when tested with dialogue transcriptions and when tested with explicit patterns of the domain). This is because our plans are focused on information of the domain, thus favoring the use of ontology over parsing.

8 CONCLUSIONS AND FUTURE WORK

Currently, several processes performed daily by humans are automated in some kind of computer system. In order to overcome the discomfort that some people feel using these types of systems and to allow the interfaces to be increasingly useful and easy to use, embodied interfaces have been developed and customized, represented by human figures or characters (LESTER et al., 2000), (GRATCH; MARSELLA, 2001), (BUISINE; ABRILIAN; MARTIN, 2004) e (GULZ; HAAKE, 2006). These interfaces interact in natural language as a conversation system. In literature these conversations systems are called by different names (dialogue systems, conversational agents, virtual humans). To Jurafsky and Martin (JURAFSJKY; MARTIN, 2009) a dialogue system has six basic components. The speech recognition component is responsible for translating the user's speech into text. The component of Natural Language Understanding produces a semantic representation suitable for the task of dialogue, usually using grammars and ontologies. The Task Manager chooses the concepts to be expressed to the user. The Natural Language Generation component chooses the concepts to be expressed to the user and defines how to express these concepts in words. The dialogue manager component controls the structure of the dialogue. The synthesizer component is responsible for translating the agents answer into speech.

In this work we realize a study of the use of NLP in NLU system for conversational agents. The study focus in the use of parsing systems to language understanding by comparing the parser impact in the task. For analysis we used an approach based on the impact of the response from the natural language understanding module for the dialogue manager module and identification of correct action to be carried out. This work contributes to state of the art in a way of reduce the lack between commercial and academic conversational agents.

8.1 Conclusions

In order to analyze the impact of parsing on conversational agents we showed the state of the art in language understanding, dialogue manager and task manager. In the natural language understanding we showed how the systems perform the understanding, as well as the elements necessary for the task. In dialogue manager we showed the elements of the conversation, development approaches and examples of their implementations. In the task manager we presented question answer systems, its elements and the tools used to identify answers.

To develop a system to evaluate the contribution of parsing systems for conversational

agents we opted for using a system of real conversation. This way we used the domain of constitutional transfer of FAMURS as a case study. In this domain there is the COMUNICA project, which is a voice QA system for Brazilian Portuguese with search capabilities for consulting both structured and unstructured datasets. Then based on this QA system we developed a conversational agent using the same resources (ontology, data set, text information and users records) to investigate what is the parsing contributions in the whole conversational process.

We also discussed the evaluation of the developed system. The evaluation concentrated both on parser and on dialogue. The parser evaluation shows the analysis of the learning and statistical process, and by human to identify the correctness of the system. In parsing evaluation, we identified that although the learning process returned a lot of rules, only with a subset of these rules the parser has able to analyze 71,5% (of 200 sentences) and for these just 55,5% returning a derivation. But, the derivations are correct in 96% (with human evaluation). In the dialogue analysis we shown the system evaluation in terms of ability to answer correctly the user's query. In this work we considered a correct answer if all steps of the pipeline presented the expected answer.

The evaluation of the parser did not resulting statistically significant difference in performance (using t-test). However, the use of a parser allows the development of BDI plans more clearly, allows the definition of relationships between beliefs, and increase the amount of information available from the NLU component.

Other possible gains include the use of the parser to identify information that might be useful to sub-systems (e.g. QA). Another benefit of the parser is the possibility of using syntactic roles (e.g. subject and object), entity recognition and synonyms and hypernyms to reduce the complexity involved in dealing a dialogue.

This work makes a contribution to the state-of-the-art reducing the difference between commercial and academic conversational agent by investigating the impact of parsing system in conversational agents pipeline. Others contributions of this work are show in (WILKENS et al., 2010a), (WILKENS et al., 2010b), (WILKENS et al., 2010c) and (WILKENS; VILLAVICENCIO, 2010).

8.2 Future Work

With the completion of this work we identify the need to expand the tests to a wider group of ratings, as well as testing with real users. Another important issue to be addressed is the operation in other domain (like travel, a very common conversational agent domain).

For the NLU module the next steps include the integration of synonym and hypernym relations to allow a better understanding of the user's needs, the use of anaphoric information (information privileged by CCG grammars). In the dialogue module, the creation of plans based on pattern extraction (using the parser) has to be accomplished, thus creating the appropriate for a new domain. The QA system also can be integrated into the conversational agent, thereby making the parser more necessary and the quality of the identification task more accurate.

In terms of parsing we identify necessity to use a system with better performance to act as the memory of the interaction with the user. In terms of grammars, we consider the use of a probabilistic grammar and the addition of grammar learning capabilities, including semantic information (translation into logical form) similar to that presented in (BOS, 2005).

APÊNDICE

Capítulo de resumo em português

A linguagem é uma marca da humanidade e da consciência, sendo a conversação (ou diálogo) uma das maneiras de comunicação mais fundamentais que aprendemos quando crianças. Por isso uma forma de fazer um computador mais atrativo para interação com usuários é usando linguagem natural. Dos sistemas com algum grau de capacidade de linguagem desenvolvidos, o chatterbot Eliza é, provavelmente, o primeiro sistema com foco em diálogo.

Com o objetivo de tornar a interação mais interessante e útil para o usuário há outras aplicações além de chatterbots, como agentes conversacionais. Estes agentes geralmente possuem, em algum grau, propriedades como: corpo (com estados cognitivos, incluindo crenças, desejos e intenções ou objetivos); incorporação interativa no mundo real ou virtual (incluindo percepções de eventos, comunicação, habilidade de manipular o mundo e comunicar com outros agentes); e comportamento similar ao humano (incluindo habilidades afetivas). Este tipo de agente tem sido chamado de diversos nomes como agentes animados ou agentes conversacionais incorporados.

Agentes conversacionais normalmente tem arquitetura com três componentes principais: ambiente (representa a interface com o mundo agente); corpo (responsável pela personificação do agente); e mente (gere os aspectos mentais). Neste trabalho nos concentramos na mente do agente, através de um sistema de diálogo, assumindo que a comunicação com o ambiente é realizado através de texto. O corpo do agente está fora do âmbito do presente trabalho.

Um sistema de diálogo possui seis componentes básicos. (1) O componente de reconhecimento de fala que é responsável por traduzir a fala do usuário em texto. (2) O componente de entendimento de linguagem natural que produz uma representação semântica adequada para diálogos, normalmente utilizando gramáticas e ontologias. (3) O gerenciador de tarefa que escolhe os conceitos a serem expressos ao usuário. (4) O componente de geração de linguagem natural que define como expressar estes conceitos em palavras. (5) O gerenciador de diálogo controla a estrutura do diálogo. (6) O sintetizador de voz é responsável por traduzir a resposta do agente em fala.

O componente de gerenciamento de tarefa utilizado nesse trabalho utiliza o modelo BDI. Esse representando uma arquitetura cognitiva para agentes inteligentes, baseada em estados mentais, e tem sua origem no modelo de raciocínio prático humano (BRATMAN; ISRAEL; POLLACK, 1988). O nome atribuído ao modelo é justificado pelos seus estados mentais:

crença, desejo e intenção (belief, desire and intention). Uma arquitetura baseada no modelo BDI representa seus processos internos através dos estados mentais acima citados, e define um mecanismo de controle que seleciona de maneira racional o curso das ações. A linguagem de programação AgentSpeak(L) foi introduzida em (??), esta é uma extensão natural da lógica de programação para a arquitetura de agentes BDI e fornece um framework para a programação de agentes BDI. Um agente AgentSpeak(L) é criado pela especificação de um conjunto de crenças em uma base e um conjunto de planos.

No entanto, não há consenso sobre os recursos necessários para desenvolver agentes conversacionais e a dificuldade envolvida nisso (especialmente em línguas com poucos recursos disponíveis). Este trabalho objetiva analisar a influência dos componentes de linguagem natural (entendimento e gerência de diálogo), analisando em especial o uso de sistemas de análise sintática (parser) como parte do desenvolvimento de agentes conversacionais com habilidades de linguagem mais flexíveis. Este trabalho analisa quais os recursos do analisador sintático contribuem para agentes conversacionais e aborda como os desenvolver, tendo como língua alvo o português (uma língua com poucos recursos disponíveis). Para isto, analisamos as abordagens de entendimento de linguagem natural e identificamos as abordagens de análise sintática que oferecem um bom desempenho. Baseados nesta análise, desenvolvemos um protótipo para avaliar o impacto do uso de analisador sintático em um agente conversacional.

Em termos de conversação agentes conversacionais são limitados pelo diálogo pré-definido em sua base de conhecimento. A fim de flexibilizar essa restrição nesse trabalho propomos o uso de sistemas de perguntas e repostas (*Question Answering* – QA). O processo de perguntas e repostas (*Question Answering* – QA) pode ser comumente dividido em quatro etapas: (1) análise pergunta, (2) a identificação de documentos candidatos, (3) geração de respostas candidatos e (4) cálculo da pontuação resposta. Estes estágios requerem as seguintes tarefas: (1) identificação do tema da questão; (2) a tradução da pergunta do usuário para uma consulta para uma ferramenta de recuperação de informação e ferramenta de pesquisa; (3) a identificação do candidato responde através do processamento dos documentos recuperados; e (4) Classificação das respostas candidatos em termos de ordem decrescente de relevância para a questão.

Um chatterbot ou agente conversacionais pode utilizar um módulo de QA para lidar com interação de linguagem natural. Nesse trabalho nós avaliamos sistemas de QA em termos de recursos utilizados: recursos rasos (por exemplo, regras baseadas em forma de superfície) e profundos (por exemplo, modelos de aprendizado de máquina); identificando uma taxa de acerto de 63.6% para métodos rasos e 69.6% para métodos profundos. Também discutimos a ligação de sistemas de QA com outras fontes de informação, que não textuais. É investigado

também a contribuição de analisadores sintáticos, em especial utilizando formalismo de Gramáticas Categoriais, para o entendimento de linguagem. Para a avaliação foram utilizados os corpora Bosque e Amazônia, onde foi identificado uma precisão de média anotação de 53.5% e uma revocação média de 74.5%.

Para analisar a contribuição do analisador sintático para agentes conversacionais desenvolvemos um *pipeline* de módulos de entendimento de linguagem e gerência de diálogo. O módulo de entendimento de linguagem realiza dois passos: análise sintática e identificação de conceitos. A análise sintática identifica a classe gramatical de cada palavra usando o analisador sintático PALAVRAS, assim como conversão da saída do analisador sintático para formato de gramáticas categoriais¹. Para a identificação de conceitos é utilizada uma ontologia de domínio para identificar palavras que representam conceitos. O módulo de gerência de diálogo utiliza as informações geradas pelo módulo de entendimento de linguagem para acessar as fontes de dados em banco de dados ou texto (através de QA). O fluxo do sistema consiste de 6 passos: entrada da pergunta, análise sintática e conversão para forma lógica, processamento BDI, geração da resposta, e apresentação da resposta.

A avaliação de diálogo mede a qualidade da resposta, por meio da análise do *pipeline* realizado internamente pelo agente, bem como a conformidade da resposta à frase de entrada. Para simular uma avaliação mais realista usamos gravações de diálogos reais, que foram transcritas e usadas como entrada para o agente. O agente com e sem o analisador sintático mostrou um desempenho semelhante na qualidade das respostas (tanto quando testado com transcrições de diálogo e quando testados com padrões explícitos do domínio). Isto devido aos planos BDI estarem focados em informações do domínio, assim favorecendo o uso de ontologias sobre análise.

Com o objetivo de avaliar o agente conversacional explicitamente no domínio utilizamos o conjuntos de padrões de perguntas. Assim como na avaliação do diálogo, avaliamos a capacidade do agente em realizar o entendimento da sentença com o parser desenvolvido, realização da ação correta e resposta adequada. Analisando os resultados observa-se que o uso de parser apresenta um resultado semelhante ao resultado sem o seu uso² No entanto o uso de analisador sintático permite a utilização de planos BDI mais específicos a tarefa a ser reali-

¹Gramática Categórica Combinatória (CCG) (STEEDMAN; BALDRIDGE, 2007) é uma forma eficiente de análise sintática baseada em um formalismo linguisticamente expressivo. Essa tem uma interface transparente entre sintaxe de superfície e representação semântica, incluindo a estrutura predicado-argumento, quantificação e estrutura de informação. Em particular, estamos utilizando nesse trabalho a implementação do analisador sintático OpenCCG.

²Resultados da avaliação: Transcrições com análise sintática (88% NLU e 70% DM), transcrições sem análise sintática (88% NLU e 70% DM), QA com análise sintática (78% NLU e 65% DM) e QA sem análise sintática (89% NLU e 68% DM).

zado, exemplo disto é o padrão utilizado “Quando <município> vai receber a primeira parcela do <transferencia constitucional> em <data>?” que em comparação com o padrão “Quanto <município> vai receber de <transferencia constitucional> em <data>?” identifica-se que o uso puramente da ontologia apresentaria falta de diferenciação entre os padrões.

De acordo com a avaliação realizada o uso de parser não apresentou uma diferença de performance estatisticamente significativa (teste T). Contudo o uso do parser permite o desenvolvimento de planos BDI de forma mais clara, assim como permite a utilização de relacionamento entre as crenças. Outro benefícios do uso de parser incluem processamento a priori da informação, que pode ser útil a sub-sistemas, por exemplo QA. Outro benefício do uso de parser é a possibilidade do uso de papéis sintáticos (exemplo: sujeito e objeto), reconhecimento de entidades e a utilização de sinônimos e hiperônimos para diminuir a complexidade envolvida no tratamento de diálogo.

Este trabalho faz uma contribuição para o estado da arte reduzindo a diferença entre agente conversacional comercial e acadêmico, através da análise do impacto de análise sintática no pipeline. Outras contribuições deste trabalho são mostradas em (WILKENS et al., 2010a), (WILKENS et al., 2010b), (WILKENS et al., 2010c) e (WILKENS; VILLAVICENCIO, 2010).

REFERÊNCIAS

- ALLEN, J. **Natural Language Understanding (2nd Edition)**. [S.l.]: Addison Wesley, 1995.
- AMARAL, C. et al. Priberam's question answering system for portuguese. In: **Evaluation of Multilingual and Multi-modal Information Retrieval - 7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006**. [S.l.: s.n.], 2006.
- ASUNCION, G.-P.; MARIANO, L.; OSCAR, C. **Ontological Engineering**. [S.l.]: Springer, 2004.
- BALDRIDGE, J. **Lexically Specified Derivational Control in Combinatory Categorical Grammar**. Thesis (PhD) — School of Informatics University of Edinburgh, 2002.
- BATES, J. **The Nature of Characters in Interactive World and The OZ**. [S.l.], 1992. CMU-CS-92-200.
- BATES, M. Encyclopedia of natural language intelligence. In: _____. [S.l.]: John Wiley Sons, 1990. chp. Natural Language Understanding.
- BATES, M. Voice communication between humans and machines. In: _____. [S.l.]: Eds. Washington, DC: National Academy Press, 1994. chp. Models of natural language understanding.
- BECERIKLI, Y.; OYSAL, Y. Modeling and prediction with a class of time delay dynamic neural networks. **Applied Soft Computing**, v. 7, p. 1164–1169, 2007.
- BICK, E. **The Parsing System “Palavras”: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework**. Thesis (PhD) — Aarhus University, 2000.
- BLACK, E. et al. A procedure for quantitatively comparing the syntactic coverage of english grammars. In: **DARPA Speech and Natural Language Workshop**. [S.l.: s.n.], 1991.
- BOS, J. Towards wide-coverage semantic interpretation. In: **Sixth International Workshop on Computational Semantics IWCS-6**. [S.l.: s.n.], 2005.
- BRATMAN, M. E.; ISRAEL, D. J.; POLLACK, M. E. Plans and resource-bounded practical reasoning. In: **Computational Intelligence**. [S.l.: s.n.], 1988.
- BROWN, E.; BARRETT, N. A wizard-of-oz platform for embodied conversational agents: Research articles. **Comput. Animat. Virtual Worlds**, 2006.
- BUISINE, S.; ABRILIAN, S.; MARTIN, J. From brows to trust. In: _____. [S.l.]: The Netherlands: Kluwer Academic Publishers, 2004. chp. Evaluation of Multimodal Behaviour of Embodied Agents.
- CASSEL, J. et al. **Embodied Conversational Agents**. [S.l.]: Cambridge, MA: MIT Press, 2000.
- CLARK, H. H.; SCHAEFER, E. F. Contributing to discourse. In: **Cognitive Science**. [S.l.: s.n.], 1989.

COHEN, P. R.; PERRAULT, C. R. Elements of a plan-based theory of speech acts. In: **Cognitive Science**. [S.l.: s.n.], 1979.

COSTA, L. First evaluation of esfinge - a question answering system for portuguese. In: **Multilingual Information Access for Text, Speech and Images, 5th Workshop of the Cross-Language Evaluation Forum, CLEF 2004**. [S.l.: s.n.], 2004.

DUIZITH, J. L. et al. A virtual assistant for websites. **Revista Eletronica de Sistemas de Informação**, v. 3, 2004.

FORNER, P. et al. Overview of the clef 2008 multilingual question answering track. In: **Francesca Borri, Alessandro Nardi e Carol Peters (eds.), Cross Language Evaluation Forum: Working Notes for the CLEF 2008 Workshop**. [S.l.: s.n.], 2008.

GIAMPICCOLO, D. et al. Overview of the clef 2007 multilingual question answering track. In: **Alessandro Nardi e Carol Peters (eds.), Working Notes for the CLEF 2007 Workshop (CLEF 2007)**. [S.l.: s.n.], 2007.

GIVÓN, T. **Syntax and Semantics Volume 12: Discourse and Syntax**. [S.l.]: New York Academic Press, 1979.

GRATCH, J.; MARSELLA, S. Tears and fears: Modelling emotions and emotional behaviors in synthetic agents. In: **International Conference on Autonomous Agents and Multiagents Systems**. [S.l.: s.n.], 2001.

GULZ, A.; HAAKE, M. Design of animated pedagogical agents: A look at their look. In: **International Journal of Human-Computer Studies**. [S.l.: s.n.], 2006.

HARABAGIU, S.; MAIORANO, S.; PASCA, M. Open-domain textual question answering techniques. In: **Natural Language Engineering 9**. [S.l.: s.n.], 2003.

ISBISTER, K.; DOYLE, P. The blind men and the elephant revisited. In: _____. [S.l.]: Kluwer Academic Publishers, 2011. chp. From brows to trust.

JULIA, H.; MARK, S. Ccgbank: A corpus of ccg derivations and dependency structures extracted from the penn treebank. **Comput. Linguist.**, 2007.

JURAFSJKY, D.; MARTIN, J. **Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics**. [S.l.]: Pearson Prentice Hall, 2009.

KAISSER, M. Qualim at trec 2005: Web-question answering with framenet. In: **TREC**. [S.l.: s.n.], 2005.

KELLEY, J. Cal – a natural language program developed with the oz paradigm: Implications for supercomputing systems. In: **First International Conference on Supercomputing Systems**. [S.l.: s.n.], 1985.

L., A. J. **How to Do Things with Words**. [S.l.]: Harvard University Press., 1962.

LEE, J. et al. Thoughts on fml: Behavior generation in the virtual human communication architecture. **The First Functional Markup Language Workshop**, 2008.

LESTER, J.; BRANTING, K.; MOTT, B. **Conversational Agents In The Practical Handbook of Internet Computing**. [S.l.]: Chapman Hall/CRC, 2004.

LESTER, J. et al. Embodied conversational agents. In: _____. [S.l.]: Cambridge: MIT Press, 2000. chp. Deictic and Emotive Communication in Animated Pedagogical Agents.

LIN, J. **Evaluation of Resources for Question Answering Evaluation**. [S.l.], 2005.

MAGNINI, B. et al. Overview of the clef 2006 multilingual question answering track. In: **Alessandro Nardi, Carol Peters e José Luís Vicedo (eds.), Cross Language Evaluation Forum: Working Notes for the CLEF 2006 Workshop (CLEF 2006)**. [S.l.: s.n.], 2006.

MOOT, R. Semi-automated extraction of a wide-coverage type-logical grammar for french. In: **Traitement Automatique des Langues Naturelles 2010, (TALN 2010)**. [S.l.: s.n.], 2010.

NAKANO M., F. K. . H. Y. . T. H. A framework for building conversational agents based on a multi-expert model. In: **9th SIGdial Workshop on Discourse and Dialogue**. [S.l.: s.n.], 2008.

NASUTO, S.; BISHOP, J.; DEMEYER, K. Communicating neurons: A connectionist spiking neuron implementation of stochastic diffusion search. **Neurocomputing**, n. 72, p. 704–712, 2009.

NYBERG, E. et al. The javelin question-answering system at trec 2002. **TREC**, 2002.

PHILLIPS, J. Generation of text from logical formulae. In: **Machine Translation**. [S.l.: s.n.], 1993.

PIERACCINI, R.; HUERTA, J. Where do we go from here? research and commercial spoken dialog systems. In: **6th SIGdial workshop on Discourse and Dialog**. [S.l.: s.n.], 2005.

POMERANTZ, J. A linguistic analysis of question taxonomies. In: **Journal of the American Society for Information Science and Technology**. [S.l.: s.n.], 2005.

POPESCU, O. **Logic-based natural language understanding in intelligent tutoring systems**. Thesis (PhD) — Carnegie Mellon University, 2005.

QUARESMA, P. et al. The university of Évora approach to qa@clef-2004. In: **Cross Language Evaluation Forum: Working Notes for the CLEF 2004 Workshop (CLEF 2004)**. [S.l.: s.n.], 2004.

QUARESMA, P. et al. The university of Évora approach to qa@clef-2004. In: **Carol Peters e Francesca Borri (eds.), Cross Language Evaluation Forum: Working Notes for the CLEF 2004 Workshop (CLEF 2004)**. [S.l.: s.n.], 2004.

QUINLAN, R. C4.5: Programs for machine learning. **Morgan Kaufmann Publishers**, 1993.

RAO, A. Agentspeak(l): Bdi agents speak out in a logical computable language. In: **Seventh Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'96)**. [S.l.: s.n.], 1996.

RAO, A.; GEORGEFF, M. P. Bdi-agents: From theory to practice. In: **First International Conference on Multiagent Systems (ICMAS'95)**. <https://www.aaai.org/Papers/ICMAS/1995/ICMAS95-042.pdf>. [S.l.: s.n.], 1995.

RICKEL, L.; JOHSON, L. Virtual humans for team training in virtual reality. In: **Ninth International Conference on Artificial Intelligence in Education**. [S.l.: s.n.], 1999.

ROJC, M. et al. Embodied conversational agents in wizard-of-oz and multimodal interaction applications. In: **Proceedings of the 2007 COST action 2102 international conference on Verbal and nonverbal communication behaviours**. [S.l.: s.n.], 2007.

RUTTKAY, Z. Situation and agency in the saiba framework, and consequences for fml. In: **The First Functional Markup Language Workshop Workshop at AAMAS**. [S.l.: s.n.], 2008.

SARMENTO, L. A first step to address biography generation as an iterative qa task. In: **Evaluation of Multilingual and Multi-modal Information Retrieval 7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006**. [S.l.: s.n.], 2006.

SCHLAEFER, N. et al. A pattern learning approach to question answering within the ephra framework. **Ninth International Conference on TEXT, SPEECH and DIALOGUE**, 2006.

SEARLE, J. R. Speech acts: Syntax and semantics volume. In: _____. [S.l.]: Academic Press, 1975. chp. Indirect speech acts.

SKANTZE, G. Galatea: a discourse modeller supporting concept-level error handling in spoken dialogue systems. In: **SIGdial6**. [S.l.: s.n.], 2005.

SRIHARI, R.; LI, W. A question answering system supported by information extraction. In: **Sixth Conference on Applied Natural Language Processing**. [S.l.: s.n.], 2000.

STEEDMAN, M.; BALDRIDGE, J. Non-transformational syntax. In: _____. [S.l.]: Blackwell, 2007. chp. Combinatory Categorical Grammar.

TRAUM, D. et al. A virtual human dialogue model for non-team interaction. In: _____. **Recent Trends in Discourse and Dialogue**. [S.l.]: Springer, 2008. chp. Recent Trends in Discourse and Dialogue.

TRUJILLO, I. **Lexicalist Machine Translation of Spatial Prepositions**. Thesis (PhD) — University of Cambridge, 1995.

TSE, D.; CURRAN, J. R. Chinese ccgbank: extracting ccg derivations from the penn chinese treebank. In: **Proceedings of the 23rd International Conference on Computational Linguistics**. [S.l.: s.n.], 2010.

USZKOREIT, H. n human language technology in human language technology. In: _____. [S.l.: s.n.], 1997. chp. Discourse and Dialogue.

VALLIN, A. et al. Overview of the clef 2005 multilingual question answering track. In: **Carol Peters (ed.), Cross Language Evaluation Forum: Working Notes for the CLEF 2005 Workshop (CLEF 2005)**. [S.l.: s.n.], 2005.

VANLEHN, K. et al. The architecture of why2-atlas: A coach for qualitative physics essay writing. In: **Intelligent Tutoring Systems**. [S.l.: s.n.], 2002.

WALLACE, R. S. **Don't read me - A.L.I.C.E. and AIML documentation**. 2011. Available from Internet: <<http://www.alicebot.com/dont.html>>.

- WALLIS, P. et al. Damian dialogue modelling for a conversational agent. In: **AI Australian Joint Conference on Artificial Intelligence**. [S.l.: s.n.], 2001.
- WARD, W.; ISSAR, S. Recent improvements in the cmu spoken language understanding system. In: **Proceedings of the workshop on Human Language Technology**. [S.l.: s.n.], 1994.
- WEIZENBAUM, J. Eliza - a computer program for the study of natural language communication between man and machine. In: **Communication of the Association for Computing Machinery 9**. [S.l.: s.n.], 1966.
- WILKENS, R.; VILLAVICENCIO, A. Question answering for portuguese: How much is needed? In: **Advances in Artificial Intelligence - SBIA 2010 - 20th Brazilian Symposium on Artificial Intelligence**. [S.l.: s.n.], 2010.
- WILKENS, R. et al. Comunica - a question answering system for brazilian portuguese. In: **International Conference on Computational Linguistics (Coling 2010)**. [S.l.: s.n.], 2010.
- WILKENS, R. et al. Comunica - a question answering system for brazilian portuguese. In: **PROPOR, International Conference on Computational Processing of the Portuguese Language**. [S.l.: s.n.], 2010.
- WILKENS, R. et al. Comunica: a voice question answering system for portuguese. In: **Proceedings of the NAACL HLT 2010 Workshop on Speech and Language Processing for Assistive Technologies**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. (SLPAT '10), p. 71–71. Available from Internet: <<http://dl.acm.org/citation.cfm?id=1867750.1867762>>.
- WOOLDRIDGE, M. J. **Reasoning about Rational Agents**. [S.l.]: MIT Press, 2000.
- ZHENG, Z. Answerbus question answering system. **HLT Human Language Technology Conference (HLT 2002)**, 2002.