



Jessica Paloma Sousa Cardoso

**Um agente conversacional pergunta-resposta
com recomendações baseadas em uma
ontologia de domínio**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática, do Departamento de Informática da PUC-Rio.

Orientador: Prof^a Simone Diniz Junqueira Barbosa

Rio de Janeiro
setembro de 2020



Jessica Paloma Sousa Cardoso

**Um agente conversacional pergunta-resposta
com recomendações baseadas em uma
ontologia de domínio**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo:

Prof^a Simone Diniz Junqueira Barbosa
Orientador
Departamento de Informática – PUC-Rio

Prof. Hélio Côrtes Vieira Lopes
Departamento de Informática – PUC-Rio

Prof. Bruno Feijó
Departamento de Informática – PUC-Rio

Rio de Janeiro, 4 de setembro de 2020

Todos os direitos reservados. A reprodução, total ou parcial do trabalho, é proibida sem a autorização da universidade, do autor e do orientador.

Jessica Paloma Sousa Cardoso

Graduou-se em Ciência da Computação (2018) pela Universidade Federal do Maranhão – Campus São Luís (UFMA)

Ficha Catalográfica

Sousa Cardoso, Jessica Paloma

Um agente conversacional pergunta-resposta com recomendações baseadas em uma ontologia de domínio / Jessica Paloma Sousa Cardoso; orientador: Simone Diniz Junqueira Barbosa. – 2020.

81 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2020.

Inclui bibliografia

1. Chatbot – Teses. 2. Interactive Question-Answering – Teses. 3. Ontologia – Teses. 4. Base de conhecimento – Teses. 5. Web Semântica – Teses. 6. Processamento de Linguagem Natural – Teses. 7. Chatbot. 8. Interactive Question-Answering. 9. Ontologia. 10. Base de conhecimento. 11. Web Semântica. 12. Processamento de Linguagem Natural.. I. Diniz Junqueira Barbosa, Simone. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

Agradecimentos

Em primeiro lugar, gostaria de agradecer a minha orientadora Prof. Simone Barbosa por ter me guiado durante esta caminhada com seu conhecimento e pelas oportunidades oferecidas a mim. Também gostaria de agradecer aos meus colegas de laboratório que contribuíram significativamente com o desenvolvimento dessa dissertação. Individualmente gostaria de agradecer aos meus colegas Dalai Ribeiro e Alysson Gomes pelo suporte técnico e esclarecimentos, e pelo fornecimento do servidor onde foi executado os experimento com os participantes.

Agradeço a meus familiares por todo apoio e confiança, em especial a minha mãe que sempre me encorajou nesse caminho. Por fim, gostaria de deixar meus agradecimentos à PUC-Rio por todo suporte fornecido.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Resumo

Sousa Cardoso, Jessica Paloma; Diniz Junqueira Barbosa, Simone. **Um agente conversacional pergunta-resposta com recomendações baseadas em uma ontologia de domínio.** Rio de Janeiro, 2020. 81p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A oferta de serviços por meio de interfaces conversacionais, ou *chatbots*, tem se tornado cada vez mais popular, com aplicações que variam de aplicativos de bancos e reserva de bilheteria a consultas em um banco de dados. No entanto, dado a quantidade massiva de dados disponível em alguns domínios, o usuário pode ter dificuldade em formular as consultas e recuperar as informações desejadas. Esta dissertação tem como objetivo investigar e avaliar o uso de recomendações na busca de informações numa base de dados de filmes através de *chatbots*. Neste trabalho, implementamos um *chatbot* por meio do uso de *frameworks* e técnicas da área de processamento de linguagem natural (NLP - *Natural Language Processing*). Para o reconhecimento de entidades e intenções, utilizamos o *framework* RASA NLU. Para a identificação das relações entre essas entidades, utilizamos as redes *Transformers*. Além disso, propomos diferentes estratégias para recomendações feitas a partir da ontologia de domínio. Para avaliação deste trabalho, conduzimos um estudo com usuários para avaliar o impacto das recomendações no uso do *chatbot* e aceitação da tecnologia por meio de um questionário baseado no *Technology Acceptance Model* (TAM). Por fim, discutimos os resultados do estudo, suas limitações e oportunidades de futuras melhorias.

Palavras-chave

Chatbot; Interactive Question-Answering; Ontologia; Base de conhecimento; Web Semântica; Processamento de Linguagem Natural.

Abstract

Sousa Cardoso, Jessica Paloma; Diniz Junqueira Barbosa, Simone (Advisor). **A question-answering conversational agent with recommendations based on a domain ontology**. Rio de Janeiro, 2020. 81p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The offer of services provided through conversational interfaces, or chatbots, has become increasingly popular, with applications that range from bank applications and ticket booking to database queries. However, given the massive amount of data available in some domains, the user may find it difficult to formulate queries and retrieve the desired information. This dissertation investigates and evaluates the use of the recommendations in the search for information on a movie database through a chatbot. In this work, we implement a chatbot with the use of frameworks and techniques from the area of natural language processing (NLP). For the recognition of entities and intents, we use the RASA NLU framework. For the identification of relations between those entities, we use the Transformers networks. In addition, we propose different strategies for the recommendation from the domain ontology. To evaluate this work, we have conducted an empirical study with volunteer users to assess the impact of the recommendations on chatbot use and the acceptance of the technology through a survey based on the Technology Acceptance Model (TAM). Lastly, we discuss the results of this study, its limitations, and avenues for future improvements.

Keywords

Chatbot; Interactive Question-Answering; Ontology; Knowledge bases; Semantic Web; Natural Language Processing.

Sumário

1	Introdução	11
2	Fundamentação Teórica	15
2.1	Web Semântica	15
2.1.1	<i>eXtensible Markup Language</i>	16
2.1.2	<i>Resource Description Framework</i>	17
2.1.3	<i>SPARQL</i>	18
2.1.4	Ontologia	19
2.2	Processamento de Linguagem Natural	20
2.2.1	<i>Named Entity Recognition</i>	21
2.2.2	<i>Relation Extraction</i>	22
2.2.3	Sistema pergunta-resposta	22
2.3	Agentes Conversacionais	23
2.4	Word Embeddings	24
3	Revisão da Literatura	26
4	Construção do Agente Conversacional	37
4.1	Sistema <i>Interactive Question Answering</i>	37
4.1.1	Processamento de Linguagem Natural	38
4.1.1.1	Pré-Processamento	38
4.1.1.2	Classificação de Intenções e Extração de Entidades	39
4.1.1.3	Extração de Relação	41
4.1.1.4	Contexto	43
4.1.1.5	Construtor de SPARQLs	43
4.2	Sistema de Recomendações por Ontologia	44
4.2.1	Busca	45
4.2.2	Embeddings	46
5	Arquitetura	49
5.1	Templates	50
5.2	Dialogs	50
5.3	Models	52
5.3.1	Comunicação com IQA e recomendações	53
5.4	Controllers	54
6	Avaliação	57
6.1	Estatísticas dos participantes	58
6.2	Estatísticas de tempo	59
6.3	Avaliação dos Bots (com e sem recomendações)	60
6.4	Avaliação das Recomendações	61
7	Conclusões e Trabalhos Futuros	64
7.1	Contribuições	64
7.2	Trabalhos futuros	64

Referências bibliográficas	66
A Questionário de avaliação	71

Lista de figuras

Figura 2.1	Exemplo de árvore XML contendo seus elementos	17
Figura 2.2	Exemplo de grafo RDF contendo recursos e suas relações	18
Figura 2.3	Árvore sintática	21
Figura 2.4	Reconhecimento de entidades nomeadas.	21
Figura 2.5	Extração de entidade e relacionamentos.	22
Figura 2.6	Exemplos de perguntas factoides e respostas que podem ser retiradas de um texto. Fonte: [14, Cap. 25]	22
Figura 2.7	Exemplos de perguntas factoides e formas lógicas utilizadas para obter respostas construídas por um <i>semantic parser</i> . Fonte: [14, Cap. 25]	23
Figura 3.1	Quantidade de trabalhos coletados no mapeamento sistemático sobre interfaces conversacionais	28
Figura 3.2	Distribuição dos trabalhos selecionados no mapeamento sistemático sobre interface conversacionais	28
Figura 4.1	Módulos do IQA.	37
Figura 4.2	Reconhecimento de entidades nomeadas.	39
Figura 4.3	Arquitetura DIET.	39
Figura 4.4	Exemplo de template utilizado no <i>Chatette</i> .	40
Figura 4.5	Reconhecimento de relações.	41
Figura 4.6	<i>Template</i> utilizado para a construção das SPARQLs.	44
Figura 4.7	Processo de construção das SPARQLs.	44
Figura 4.8	Árvores representando as hierarquias na ontologia.	45
Figura 4.9	Árvores de gêneros.	46
Figura 4.10	Árvores de recomendação.	47
Figura 4.11	Ordenação aleatória.	47
Figura 4.12	Ordenação por similaridade dos <i>embeddings</i> .	48
Figura 5.1	Visão geral dos componentes do <i>chatbot</i> .	49
Figura 5.2	Detalhes das classes do módulo <i>Dialogs</i> .	52
Figura 5.3	Mensagem formatada para paginação de acordo com <i>templates.ini</i>	52
Figura 5.4	Detalhes das classes do módulo <i>Models</i> .	53
Figura 5.5	Detalhes das classes do submódulo <i>Ontoexplorer</i> .	55
Figura 5.6	Detalhes das classes do módulo <i>Controllers</i> .	56
Figura 5.7	Fluxo da conversa com IQA	56
Figura 6.2	Tempo gasto pelos usuários em cada bot	60
Figura 6.3	Quantidade de mensagem por bot	61
Figura 6.4	Questionário dos bots com recomendação	62
Figura 6.5	Questionário dos bots sem recomendação	62
Figura 6.6	Avaliação das recomendações por bot.	63

Lista de tabelas

Tabela 2.1	Tabela representando as propriedades com seus <i>domains</i> e <i>ranges</i>	18
Tabela 3.1	Resumo dos trabalhos da literatura	30
Tabela 6.1	Grupos do experimento	57

1 Introdução

Nos dias de hoje, praticamente todo tipo de serviço que utilizamos é conectado à Internet. Alguns exemplos são serviços de *streaming*, compras online, redes sociais, sistemas de cadastro para emissão de documentos etc. A quantidade de usuários na Internet aumentou dez vezes entre 1999 e 2013¹. Segundo relatório divulgado *Hootsuite* e *We are social*,² em 2018, o número de pessoas conectadas correspondia a mais de 50% da população mundial. Além disso, comparado ao ano anterior, a maior parte dos novos usuários se conectaram através de *smartphones* e planos de dados móveis. Como consequência disso, uma grande quantidade de informações é disponibilizada diariamente por seus usuários via *smartphones*, *tablets*, *desktops* e outros dispositivos que possuam acesso a rede mundial de computadores.

Conseqüentemente, esses dados podem ser utilizados por aplicações e serem apresentados de acordo com algum critério, como ocorre com o *PageRank* [2, 3], o qual atribui um nível de relevância para cada página da Web. Essa medida pode ser utilizada em sistemas de buscas para ordenar os resultados. Os dados também podem ser utilizados em vários aplicativos e serviços que façam uso de sistemas de recomendação e recuperação de informação. Por exemplo, um sistema de recomendação pode sugerir ao usuário um conjunto de vídeos com base no seu comportamento, fazer uma sugestão de amizade ou propor um produto baseado em compras efetuadas anteriormente.

Esses sistemas estão disponíveis através da *Web*, a qual é representada como uma rede de documentos interligados através de links, onde cada um deles está associado a um URL³ que permite a sua localização e acesso na rede. A estrutura das páginas *Web* é descrita através da linguagem de marcação HTML⁴, utilizada para criar documentos que contêm textos, links, *scripts* e elementos multimídia, como imagens, vídeos e fotos.

Os documentos são acessados através de clientes que implementem o protocolo HTTP (*Hypertext Transfer Protocol*). O cliente envia uma requisição ou solicitação a um servidor, que disponibiliza o recurso requisitado (ex.: página

¹<https://www.internetlivestats.com/internet-users/>

²<https://wearesocial.com/blog/2018/01/global-digital-report-2018>

³<https://www.w3.org/Addressing/URL/url-spec.html>

⁴<https://www.w3.org/standards/webdesign/htmlcss>

HTML). O processo de recuperação dos documentos em *engines* de busca geralmente é feito através de algoritmos de *ranking*, onde são utilizadas técnicas de extração de informações baseadas em conteúdo ou busca estatística, como o *PageRank*.

O número de usuários presentes na Internet teve um aumento significativo e, como consequência disso, uma grande quantidade de informações foram disponibilizadas na rede sem qualquer tipo de controle ou descrição desses dados. Souza et al. (2004) [4] discutem o cenário da *Web* diante dessas circunstâncias:

Embora tenha sido projetada para possibilitar o fácil acesso, intercâmbio e a recuperação de informações, a *Web* foi implementada de forma descentralizada e quase anárquica; cresceu de maneira exponencial e caótica e se apresenta hoje como um imenso repositório de documentos que deixa muito a desejar quando precisamos recuperar aquilo de que temos necessidade.

De acordo com os autores, há uma limitação quando se deseja recuperar ou consultar esses dados, visto que a ausência de informações descritivas e metadados dificulta o processo de interpretação e compreensão dos dados por sistemas computacionais como mecanismos de buscas. Assim, com o propósito de facilitar o processo de recuperação dos dados por máquinas, Berners-Lee et al. (2001) [5] propuseram a *Web Semântica*, que consiste em uma extensão da *Web* tradicional onde os dados são bem definidos, permitindo uma melhor cooperação entre pessoas e máquinas. Foram desenvolvidas diversas tecnologias, como *eXtensible Markup Language* (XML), *Resource Description Framework* (RDF), *Web Ontology Language* (OWL), que são utilizadas com propósito de incorporar semântica aos dados. Elas nos permitem trazer mais significados sobre os dados, por exemplo, possibilitando-nos realizar inferências sobre eles.

Os dados na *Web semântica* geralmente são representados através de RDF e a linguagem de consulta SPARQL nos permite recuperar esses dados de acordo com o conjunto de instruções repassadas. Entretanto, o acesso a esses metadados exige conhecimento especialista sobre as tecnologias de *Web Semântica*, tendo como consequência uma quantidade limitada de pessoas que conseguem utilizar essas informações.

Segundo o relatório divulgado por *Hootsuite* e *We are social*,⁵ no ano de 2018, 52% da população utilizou dispositivos móveis, seguidos por *desktops* (43%), *tablets* (4%) e outros dispositivos (0,14%). Dentre esses dispositivos,

⁵<https://wearesocial.com/blog/2018/01/global-digital-report-2018>

os *desktops* e os *tablets* estiveram em queda nos últimos anos, diferente dos *smartphones* e outros dispositivos que continuam em crescimento. Os usuários gastam 75%⁶ do tempo em aplicativos; dentre esses, metade do tempo é utilizado em aplicativos de redes sociais e outras formas de comunicação como mensageiros instantâneos e e-mail.

Dada a popularidade dos aplicativos de mensagens instantâneas, muitos deles integraram a funcionalidade de *chatbots*: o *WhatsApp*, o *Slack*, o *Telegram* e o *Messenger* são alguns exemplos. Um *chatbot* é um agente conversacional que possui como objetivo a troca de informação através de diálogos, assemelhando-se a uma conversa humana. Chatbots são comumente utilizados em serviços de *e-commerce* com intuito de agilizar o processo de atendimento ao cliente, como preenchimento de dados que podem ser úteis para o fornecimento de informações sobre algum produto ou serviço. A integração desses serviços em aplicativos populares de mensagens traz maior praticidade ao usuário, ao possibilitar a realização de operações, consultas ou serviços sem a necessidade de instalar um novo aplicativo.

Em suma, com a fonte de dados massiva da web, tecnologias da Web semântica, novas formas de captura de dados por meio de smartphones e outros periféricos, e recentes avanços na área de *Big Data*, *Machine Learning* e *Deep Learning*, foram possíveis avanços consideráveis e um renovado interesse em aplicações que fazem uso de *Natural Language Processing* (NLP), como é o caso dos chatbots. Unindo essas tecnologias, propomos neste trabalho a construção de um agente conversacional que utiliza uma ontologia para fornecer recomendações e permite que o usuário utilize o conhecimento capturado nessa ontologia para formulação de novas consultas. A nossa principal contribuição está focada na estratégia utilizada em nossa recomendação, que tem como essência o uso de uma ontologia.

Este documento encontra-se organizado em seis capítulos, desconsiderando o capítulo de Introdução, sendo eles:

- **Capítulo 2** – neste capítulo apresentamos conceitos fundamentais para o desenvolvido desta dissertação, tais como web semântica e processamento de linguagem natural.
- **Capítulo 3** – realizamos uma revisão da literatura sobre agentes conversacionais que façam uso de base de conhecimento.

⁶<https://bit.ly/mobile-games-now-account-for-33-of-installs>

- **Capítulo 4** – neste capítulo descrevemos os passos do desenvolvimento do nosso agente conversacional e a estratégia de recomendação baseada na ontologia.
- **Capítulo 5** – detalhamos a arquitetura desenvolvida para o nosso agente conversacional.
- **Capítulo 6** – os resultados e estratégias de avaliação da nossa metodologia são descritos neste capítulo.
- **Capítulo 7** – no último capítulo deste documento apresentamos as considerações finais sobre a metodologia proposta e também comentamos possíveis trabalhos futuros.

2 Fundamentação Teórica

Para o desenvolvimento deste trabalho, necessitamos primeiramente definir um conjunto de conceitos essenciais. Na Seção 2.1 descrevemos conceito de *Web Semântica* com suas tecnologias e a importância delas na tarefa de organização das informações da *Web*. Em seguida, na Seção 2.2 discutimos sobre processamento de linguagem natural, destacando as áreas de sistema pergunta-resposta, reconhecimento de entidade nomeada e extração de relação. Por fim, na Seção 2.3 mostramos os principais tipos e diferenças entre *chatbots*, bem como seus conceitos e definições.

2.1 Web Semântica

Desde a sua criação, a quantidade de dados disponível tem aumentado significativamente a cada dia e, atualmente, existem mais de um bilhão de *websites* e mais de quatro bilhões de usuários na internet. O conceito de *Web* foi proposto por Tim Berners-Lee e Robert Cailliau¹ em 1989, como uma rede (*web*) de documentos de hipertextos, onde cada página representa um documento *Web*, escrito em HTML. Esses documentos HTML podem conter links para outras páginas e ser divididos por tags HTML que contêm textos, *scripts*, elementos multimídia etc., e podem ser acessados através de navegadores que implementam o protocolo HTTP.

Entretanto, esses dados encontram-se organizados para permitir uma melhor utilização por pessoas, e não por agentes de software, visto que a *web* de documentos não foi construída dando ênfase aos dados. Assim, em 2001 foi proposta a *Web* semântica, que correspondia a uma extensão da tradicional *Web* de documentos.

A *Web* semântica organiza as informações das páginas *Web*, permitindo a interação através de agentes de software que possam auxiliar o usuário no processo de recuperação desses dados, correspondendo à *Web* dos Dados Conectados, de acordo com o W3C².

¹<https://www.w3.org/History/1989/proposal.html>

²<http://www.w3c.br/Padroes/WebSemantica>

Assim, é necessário que os dados estejam disponíveis através de padrões preconcebidos por meio de tecnologias da *Web* semântica, também conhecida como *Web* dos Dados Conectados. Esses dados devem possuir relacionamentos semânticos entre si, diferente de uma abordagem tradicional de dados onde essa relação não é necessária. Portanto, foi criado um conjunto de tecnologias que têm como intuito organizar e fornecer um significado a dados, algumas das principais sendo *Resource Description Framework* (RDF), *eXtensible Markup Language* (XML), *SPARQL Protocol* e *RDF Query Language*³, *Web Ontology Language* (OWL) e outras ferramentas que são definidas pelo *World Wide Web Consortium* (W3C).

2.1.1

eXtensible Markup Language

Extensible Markup Language, abreviado como XML, é uma linguagem de marcação. Diferente do HTML, ela permite a construção de documentos utilizando estruturas arbitrárias. Segundo o W3C, alguns dos principais objetivos do XML são:

- Facilidade em escrever programas para processar documentos XML
- Formato formal e conciso
- Facilidade em criar documentos XML
- Legível para a compreensão humana apresentando clareza

Essa linguagem permite a criação livre de *tags*, dessa forma fornecendo maior riqueza na descrição do dados. Ela possui *elements*, que correspondem a entidades e são definidas similarmente ao HTML, representadas como `<filme>` e `</filme>`, por exemplo. Entre essas tags pode haver texto ou elementos, assim permitindo um conjunto aninhado de *tags*. Quando esse conjunto encontra-se sob uma *tag* raiz, diz-se que esse documento está bem formado.

Na Figura 2.1 ilustramos a árvore XML formada a partir dos elementos contidos no XML exemplificado na figura. Percebemos a hierarquia formada entre os elementos, onde os estudantes (*tag* `<estudante>`) fazem parte do elemento raiz (*tag* `<escola>`).

³<http://www.w3c.br/Padroes/WebSemantica>

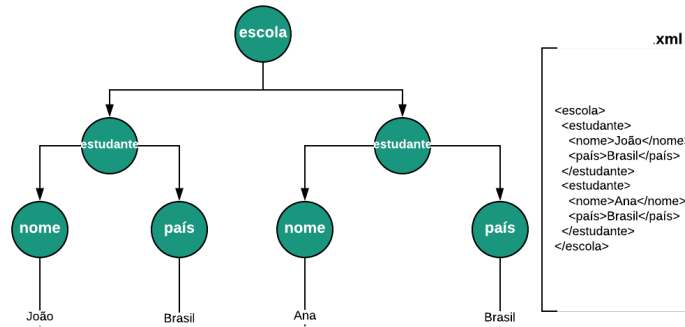


Figura 2.1: Exemplo de árvore XML contendo seus elementos

2.1.2

Resource Description Framework

O *Resource Description Framework* (RDF) possui um *graph-based model* diferente do XML, que tem um *tree-based model*. O RDF também é um padrão adotado pelo W3C, que representa informações sobre os recursos. Ele estrutura os dados para que esses possam ser processados por máquinas ao relacionar recursos. O **sujeito** e o **objeto** representam os recursos, e a relação entre eles é definida por um **predicado**. Esses termos definem uma **tripla** RDF. A seguir mostramos alguns exemplos de triplas RDF em pseudocódigo:

- <João> <é amigo de> <Ana>
- <João> <nasceu em> <Brasília>
- <João> <é irmão de> <Maria>
- <Brasília> <é a capital do> <Brasil>
- <Ana> <mora no> <Brasil>
- <Ana> <tem um> <cachorro>

Nos exemplos acima, notamos que um recurso pode ser reutilizado em outras relações como sujeito ou objeto, permitindo a conexão entre os dados. O sujeito e predicado possuem um identificador universal chamado IRI (*International Resource Identifier*), que diferencia um recurso do outro. O predicado na última relação é representado como um literal, diferente das demais triplas que possuem IRI.

Na Figura 2.2 ilustramos um exemplo de grafo RDF montado a partir dos exemplos mencionados anteriormente.

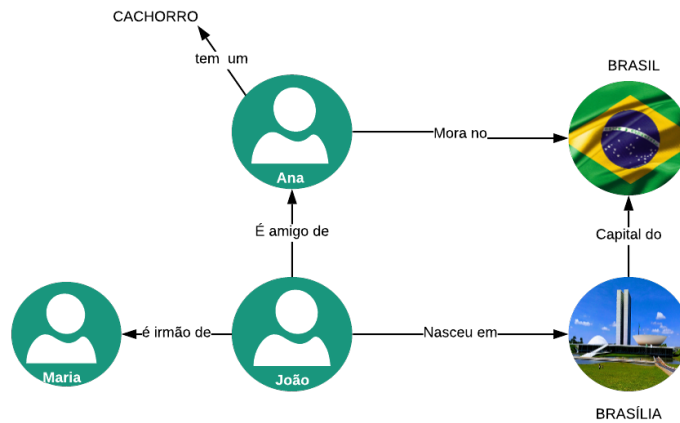


Figura 2.2: Exemplo de grafo RDF contendo recursos e suas relações

Com intuito de facilitar a interpretação das triplas RDF por máquinas, o RDF faz uso do XML. Além disso é possível incorporar um vocabulário que é nomeado *RDF Schema*. São utilizadas *Uniform Resource Identifier (URI)* para identificar os sujeitos, predicados e objetos, sendo que os objetos também podem ser representados como literais. No Esquema RDF são utilizadas classes para representar os recursos RDF; por exemplo, na Figura 2.2 João, Maria e Ana podem ser modelados como pertencem a uma classe Pessoa. O relacionamento entre esses recursos são definidos através de propriedades, as quais possuem restrições a cerca dos recursos, sendo elas: *domain* – define a qual classe pertence a propriedade – e *range* – que informa qual é tipo do objeto da tripla. Construímos a Tabela 2.1 para representar os relacionamentos da Figura 2.2.

Tabela 2.1: Tabela representando as propriedades com seus *domains* e *ranges*

<i>domain</i>	<i>property</i>	<i>range</i>
Pessoa	é amigo de	Pessoa
Pessoa	nasceu em	Cidade
Pessoa	é irmão de	Cidade
Cidade	é a capital do	Estado/País
Pessoa	mora em	Cidade/Estado/País
Pessoa	tem um	Pet

2.1.3 SPARQL

Outra tecnologia utilizada na *Web Semântica* é o protocolo SPARQL, que lida com a tarefa de recuperação e manipulações do conjunto de dados

RDF. As consultas SPARQL podem ser do tipo: **SELECT**, que retorna um conjunto de dados de acordo com o padrão especificado na consulta; **DESCRIBE**, que retorna o grafo RDF que representa os dados sobre os recursos; **CONSTRUCT**, que realiza a recuperação de um grafo RDF, em alternativa ao SELECT, que só retorna valores; **ASK**, que fornece um resultado lógico ao verificar se o padrão especificado possui ou não um resultado no *dataset*.

Uma consulta SPARQL pode conter *prefix declarations*, *dataset definition*, *result clause* e *query modifiers*, como representado a seguir:⁴

```

1 # prefix declarations
2 # nomeia uma URI de forma a abreviar o link
3 PREFIX foo: <http://example.com/resources/>
4 ...
5 # dataset definition
6 # Indica os grafos RDFS consultados
7 FROM ...
8 # result clause
9 # Indica a informação a ser retornada da consulta
10 SELECT ...
11 # query pattern
12 # Especifica o que consultar no conjunto de dados
13 WHERE {
14     ...
15 }
16 # query modifiers
17 # reorganiza os dados da consulta
18 # pode limitar a quantidade de resultados, reordenar, etc.
19 ORDER BY ...

```

2.1.4 Ontologia

O termo ontologia é originário do grego a partir da junção dos termos *onto*, que significa ser, e *logos*, que significa palavra. Na área de computação, uma ontologia é vista como uma forma de definir relações entre os dados. Existem diversas definições para ontologia.

⁴<https://www.w3.org/2009/Talks/0615-qbe/>

Segundo Gruber (1995) [7], uma ontologia pode ser descrita como uma especificação de conceitualizações, assim ela descreve os conceitos e os relacionamentos entre eles. Ainda de acordo com as definições de Breitman (2000) [8] e Gruber (1995) [7], podemos afirmar que as ontologias têm como propósito facilitar o processo de descrição e recuperação de dados por agentes de software, sendo uma ferramenta importante na Web Semântica, que visa construir uma Web de Dados.

As ontologias podem ser classificadas de acordo com a sua função em cinco categorias: (1) Ontologia Genérica, que realiza a descrição geral de conceitos não limitados a um domínio específico; (2) Ontologia de Domínio, que, diferente da genérica, é limitada a um domínio particular de conhecimento; (3) Ontologias de Tarefas, que focam na descrição de atividades ou tarefas, e também são independentes de domínio; (4) Ontologias de aplicação, que utilizam os termos das ontologias (2) e (3) para definir regras na aplicação de tarefas pelas entidades do domínio; e, por fim, (5) Ontologias de representação, que clarificam os conceitos e formalismos que representam o conhecimento [6].

As ontologias são comumente especificadas como uma 4-tupla contendo classes, relacionamentos, instâncias e axiomas [9], sendo representadas como:

$$O = \{C, R, I, A\}$$

onde C corresponde a classes, R relacionamentos, I um conjunto de instâncias das classes, R relacionamentos entre os conceitos/classes e A os axiomas expressos através de uma linguagem lógica [6].

Por fim, é importante citar uma das linguagens mais populares para descrever ontologias, que é a *Web Ontology Language* (OWL)⁵. Proposta pelo W3C, permite uma melhor descrição de conteúdo semântico de dados em XML, RDF, etc. Também possui suporte por diversos softwares, como Protégé⁶, sendo bastante difundida no meio acadêmico.

2.2

Processamento de Linguagem Natural

O processamento de linguagem natural (PLN) corresponde a uma subárea da inteligência artificial (IA) que é responsável pela compreensão e geração de textos em idiomas humanos, sejam eles orais ou escritos. Existem diversas aplicações onde o PLN é utilizado, tais como sistemas de recuperação de informação, tradução por máquina (*machine translator*), pergunta-resposta, extração de informação e sistemas de diálogos [18].

⁵<https://www.w3.org/OWL/>

⁶<https://protege.stanford.edu/>

Os estudos em processamento de linguagem natural podem ser classificados de acordo com sua análise em cinco níveis: sintático, semântico, morfológico, fonético e pragmático [10]. A sintaxe lida com a estrutura da frase (Figura 2.3), a semântica com o significado da sentença, a morfologia estuda a formação das palavras, a fonética envolve a pronúncia das estruturas sintáticas e, por fim, a pragmática verifica se o significado é o mais apropriado àquela estrutura [11].

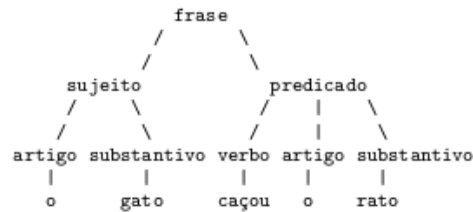


Figura 2.3: Árvore sintática

Como a área de PLN possui diversos campos de estudos, é importante frisar aqueles de maior importância nesta dissertação. Esses tópicos serão abordados a seguir.

2.2.1

Named Entity Recognition

Com o objetivo de identificar nomes de pessoas, organização e localizações geográficas, foi definido o termo Entidade Nomeada (NE, do inglês *Named Entity*) na sexta *Message Understanding Conference* (MUC-6) [19]. Uma entidade nomeada é uma forma de categorizar um elemento como pertencente a um grupo de elementos com atributos similares a ele, e desse modo, podemos considerar como entidades data, quantidade, porcentagens, tempo e assim por diante.

Assim, a tarefa reconhecimento de entidade nomeada (NER, do inglês *Named Entity Recognition*) tem como objetivo identificar os elementos de interesse do texto e marcá-los de acordo com as categorias que correspondem às entidades nomeadas. Na Figura 2.4, consideramos as entidades filme, data e localização e destacamos os termos nas frases que são reconhecidos como pertencentes às categorias correspondentes.

O filme **Procurando Nemo** estreou em **05/12/2003** no **Brasil**.

Figura 2.4: Reconhecimento de entidades nomeadas.

2.2.2

Relation Extraction

Segundo Bach et al. (2007) [20], uma relação pode ser definida como uma tupla de entidades, onde as relações são predefinidas por um dado documento. No geral, a maioria das aplicações consideram relações binárias. Na Figura 2.5 mostramos um exemplo onde são extraídas as relações binárias *gravado_em*(Filme, Localização) e *dirigido_por*(Filme, Pessoa).



Figura 2.5: Extração de entidade e relacionamentos.

2.2.3

Sistema pergunta-resposta

Question Answering (QA) representa uma subárea de processamento de linguagem natural relacionada à recuperação de informação. Tem como objetivo responder uma pergunta em linguagem natural escrita pelo usuário e recuperar uma informação ou conjunto de informações relevantes a partir de um ou vários conjuntos de dados.

A maioria dos sistemas pergunta-resposta são construídos com o objetivo de responder perguntas que possuem suas respostas expressas em pequenos trechos; essas perguntas são chamadas de *factoid questions*. Os dois principais paradigmas utilizados para a resolução desse problema são: *Information Retrieval-based question answering*, que recupera a resposta a partir de passagens de texto usando algoritmos de *reading comprehension* (Figura 2.6); e *Knowledge-based Question Answering*, que relaciona a pergunta a uma representação lógica (Figura 2.7) [14, Cap. 25].

Question	Answer
Where is the Louvre Museum located?	in Paris, France
What's the abbreviation for limited partnership?	L.P.
What are the names of Odin's ravens?	Huginn and Muninn
What currency is used in China?	the yuan
What kind of nuts are used in marzipan?	almonds
What instrument does Max Roach play?	drums
What's the official language of Algeria?	Arabic
How many pounds are there in a stone?	14

Figura 2.6: Exemplos de perguntas factoides e respostas que podem ser retiradas de um texto. Fonte: [14, Cap. 25]

Question	Logical form
When was Ada Lovelace born?	birth-year (Ada Lovelace, ?x)
What states border Texas?	$\lambda x. \text{state}(x) \wedge \text{borders}(x, \text{texas})$
What is the largest state	$\text{argmax}(\lambda x. \text{state}(x), \lambda x. \text{size}(x))$
How many people survived the sinking of the Titanic	$(\text{count} (\text{!fb:event.disaster.survivors fb:en.sinking.of.the.titanic}))$

Figura 2.7: Exemplos de perguntas factoides e formas lógicas utilizadas para obter respostas construídas por um *semantic parser*. Fonte: [14, Cap. 25]

Allam et al. (2012) [21] dividem o sistema QA em três grandes módulos, sendo eles a classificação da pergunta, a recuperação da informação e a extração da resposta. Na primeira etapa há a categorização do tipo da pergunta; a recuperação da informação tem como objetivo coletar elementos importantes de acordo com os termos identificados na pergunta; e, por fim, temos o módulo que extrai a resposta em conformidade com a busca do usuário.

O domínio de um sistema QA pode ser fechado, ou seja, limitado a responder perguntas sobre um domínio específico, ou aberto, onde podem ser realizadas perguntas sobre “qualquer coisa” [22]. Além disso, os autores classificam os tipos de questões em oito categorias, sendo elas: (1) factóide (quem, quando, onde) - a resposta requer um fato único; (2) confirmação (sim, não) - respostas que podem ser sim ou não; (3) definição - respostas que representam a definição de alguma coisa; (4) causal (como, por quê, qual) - respostas que podem representar a consequência de um ou vários fatos; (5) procedimental - responde com etapas utilizadas para realizar algo; (6) comparativa - responde uma comparação entre dois ou mais elementos; (7) com exemplos - resposta com exemplos que descrevem o que foi perguntado; e (8) Opinião - a resposta é uma opinião sobre algo.

Gao et al. (2019) [15] definem dois tipos de sistemas *question-answering* de acordo com tipo de dado: *Knowledge Base Question Answering* (KB-QA), que utiliza bases de conhecimento como a *DBPedia* [12], *YAGO* [23] e *Wikidata* [13]; e *Text-QA*, que extrai a resposta a partir de uma coleção de documentos. Além disso, o QA ainda pode ser classificado como turno simples, onde o usuário envia tudo o que deseja em uma única consulta, ou multi-turno (*Conversational QA*), que possibilita ao usuário decompor a sua pergunta em vários turnos.

2.3

Agentes Conversacionais

Os agentes conversacionais (AC), também conhecidos como sistemas de diálogos, são *softwares* que se comunicam através de textos escritos e/ou falados. Estes tipos de software analisam os textos em linguagem natural e

respondem de acordo com tipo de tarefa para qual foram programados. De acordo com Gao et al. (2019) [15], os tipos de problemas tratados por sistemas de diálogos podem ser divididos em três categorias: *question answering*, que visam a recuperar informações de uma base a partir de consultas realizadas em linguagem natural; *task completion*, que têm como objetivo e cumprir tarefa, por exemplo, reserva de passagem aérea; e, por fim, *social chat*, cujo propósito é conversar com o usuário de forma similar a um diálogo humano-humano.

Um sistema conversacional pode ser classificado em dois tipos: *Task-Oriented Systems*, que auxilia o usuário na execução de tarefa, podendo ser a tarefa de responder uma pergunta ou agendar compromissos; e *Non-Task-oriented Systems*, que lida com o problema de *chitchat* [16].

Um *Task-Oriented System* cuja tarefa seja recuperar informações a partir de textos em linguagem natural também pode ser visto como um sistema de *Interactive Question Answering* (IQA). Segundo Konstantinova et al. (2013) [24], um IQA possibilita a composição de perguntas de acordo com o resultado. Esses sistemas também podem permitir o uso de *follow-up questions*, o que pode ajudar a obter outras informações sobre a resposta ou sobre a pergunta original.

2.4 Word Embeddings

Word embeddings são representações em forma de vetores multidimensionais reais das palavras, localizadas em um espaço vetorial pré-definido, no qual a geometria entre esses vetores captura as relações semânticas entre as palavras correspondentes. *Word embeddings* são geralmente utilizados com a similaridade de cosseno, dado pela fórmula:

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}} \quad (2-1)$$

O valor resultante fica no intervalo $[-1, 1]$, que significa:

- 1: Os vetores são paralelos e de mesmo sentido
- 0: Os vetores são perpendiculares
- -1: Os vetores são paralelos e com sentidos opostos

Palavras com contexto similar ocupam posições espaciais mais próximas. Por exemplo, palavras como ‘homem’ e ‘mulher’ terão um valor de similaridade próximo ao de ‘rei’ e ‘rainha’.

O próximo capítulo apresenta uma breve revisão da literatura de interfaces conversacionais.

3

Revisão da Literatura

Para termos uma ideia de quais abordagens na construção de interfaces conversacionais estão apresentando os melhores resultados atualmente, decidimos realizar um mapeamento sistemático de trabalhos publicados recentemente, mais especificamente os trabalhos publicados em conferências e periódicos a partir do ano de 2017. Para direcionar ainda mais nossa busca por artigos que possam contribuir com a nossa pesquisa, estabelecemos a seguinte pergunta, que deveria ser respondida a partir dos trabalhos coletados: Qual é o estado da arte na construção de interfaces conversacionais que incorporam conhecimento de domínio específico?

Para a seleção dos trabalhos que poderiam responder nossa questão de pesquisa, seguimos a abordagem de coleta e filtragem de artigos baseados em um mapeamento sistemático da literatura. Definimos os seguintes passos para execução desse mapeamento:

1. Coletar trabalhos buscando por termos contidos em seu títulos, palavras-chaves ou resumos;
2. A partir da leitura dos metadados do trabalho e seu resumo, filtrar os trabalhos seguindo nossos critérios de exclusão:
 - São artigos duplicados;
 - Não estão em inglês ou português;
 - Não são *full papers* publicados em conferências ou periódicos;
 - Não têm contribuição clara ou sua contribuição não está relacionada com a nossa questão de pesquisa;
 - Apresentam apenas opiniões, sem indícios de terem dados que suportem suas afirmações.
3. Ler introdução e conclusão dos trabalhos selecionados, agrupando-os com base na sua principal contribuição e realizar nova filtragem aplicando os critérios de exclusão, com a adição de mais um item nestes critérios:
 - Não temos acesso ao texto completo.
4. Ler integralmente os trabalhos selecionados e escrever um resumo a respeito de cada um;

5. Realizar “*snowball search*” (a partir de um artigo, coletar trabalhos que ele cita ou outros trabalhos onde tal artigo é citado) em alguns artigos selecionados e aplicar os passos 2, 3 e 4 nos trabalhos coletados nesta busca;
6. Reportar os resultados obtidos.

Para a execução do passo 1 do nosso mapeamento sistemático buscamos trabalhos indexados nas bases de dados da Scopus, ACM e arXiv. Cada uma dessas bases contém uma forma particular de construção das *queries* de busca, entretanto a *query* de busca que executamos pode ser descrita genericamente da seguinte forma:

- campos_de_busca=título ∨ palavras-chave ∨ resumo, ano_publicacao ≥ 2017, buscar_por=(conversational interface ∨ conversational agent ∨ chatbot ∨ dialogue ∨ question answering) ∧ (ontology ∨ domain knowledge)

A *query* acima foi construída com o intuito de: sermos bem específicos no que diz respeito a coletarmos apenas trabalhos que incorporassem algum conhecimento sobre interfaces conversacionais e conseguirmos coletar trabalhos que, mesmo não tratando diretamente de interfaces conversacionais, poderiam nos ajudar a descobrir novas abordagens para responder perguntas feitas em linguagem natural.

A execução da busca em todas as bases de dados citadas anteriormente resultou na coleta de 337 trabalhos na Scopus, 67 na ACM e 63 na arXiv, somando-se um total de 467 trabalhos. Porém, em alguns casos um mesmo trabalho foi retornado por bases de dados diferentes, como pode ser visto na Figura 3.1. Removendo-se essas duplicidades, nossa busca retornou um total de 405 trabalhos.

Após a coleta, precisamos filtrar os trabalhos segundo os critérios de exclusão que estabelecemos; logo, executamos os passos 2 e 3 do nosso mapeamento sistemático. Devemos citar que alguns trabalhos contidos na base do arXiv ainda não haviam passado por revisão, e conseqüentemente não tinham sido publicados. Por isso, esses trabalhos foram removidos da nossa análise. Concluído este processo, selecionamos um total de 46 trabalhos. Agrupamos os trabalhos selecionados segundo suas contribuições em 5 categorias:

- *Approach* - A contribuição é uma técnica, metodologia, nova arquitetura de rede neural, etc;
- *Tool* - A contribuição é uma ferramenta, sistema, biblioteca, etc;
- *Review* - A contribuição é uma revisão da literatura sobre o tema;

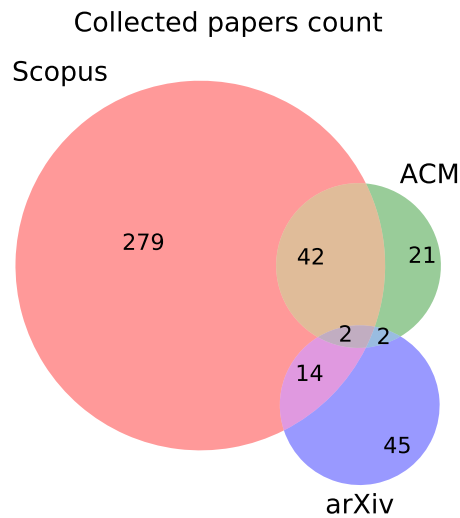


Figura 3.1: Quantidade de trabalhos coletados no mapeamento sistemático sobre interfaces conversacionais

- *Resource* - A contribuição é um recurso qualquer como uma ontologia, *dataset*, etc;
- *Other* - A contribuição não se encaixa em nenhuma das categorias anteriores e precisa ser investigada mais a fundo posteriormente, lendo-se o texto completo, para criação de uma nova categoria ou associação a uma das existentes.

O resultado final da seleção e categorização dos artigos pode ser vista na Figura 3.2.

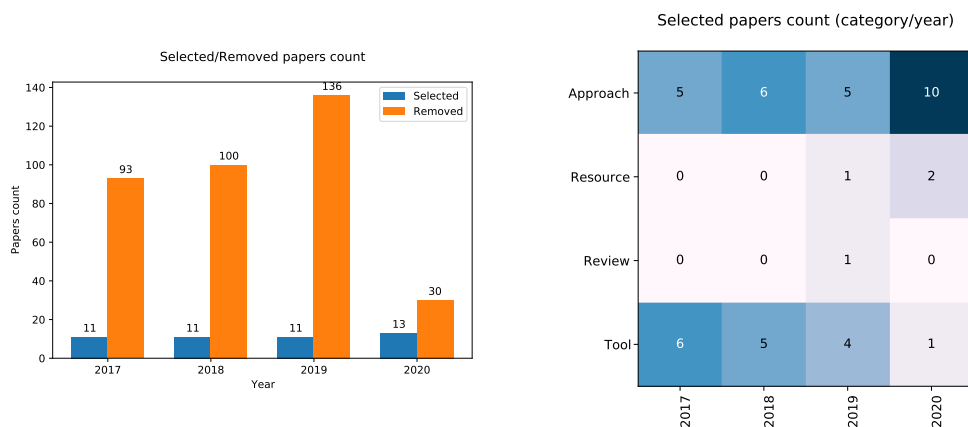


Figura 3.2: Distribuição dos trabalhos selecionados no mapeamento sistemático sobre interface conversacionais

Após selecionados os trabalhos, executamos o passo 4 do nosso mapeamento sistemático, ou seja, fizemos uma leitura integral e sumarização deles. Decidimos começar pela leitura completa dos trabalhos classificados como *Tool*,

e buscamos responder as seguintes perguntas durante nossa leitura e sumariação:

- Qual o propósito da ferramenta?
- Quais os idiomas suportados por ela?
- Como a ferramenta detecta a intenção no diálogo?
- Como ela lida com o estado e contexto da conversa?
- Como a ferramenta foi avaliada?
- Como incorpora conhecimento de domínio no diálogo?

Na tabela 3.1 resumimos os resultados obtidos, detalhados a seguir.

Tabela 3.1: Resumo dos trabalhos da literatura

Mapeamento Sistemático						
Estudo	Propósito	Idioma	Intenção	contexto	Avaliação	Domínio
Qin et al. (2020) [25]	Basquete	chinês	similaridade	N/A	avalia entidades apenas	grafo de conhecimento
Kulkarni et al. (2019) [27]	produtos	inglês	similaridade questões-respostas	N/A	<i>dataset</i> de teste	004
Kadariya et al. (2019) [28]	Pacientes Asma	inglês	<i>DialogFlow</i>	<i>NoSQL</i>	com usuários	<i>NoSQL</i>
Avila et al. (2019) [29]	Medicamentos	Português	palavra-chave	<i>NoSQL</i>	com usuários	ontologia
Mittal et al. (2018) [30]	interface conversacional	inglês	N/A	pelo usuário	<i>dataset</i> de teste	ontologia
Le-Hong et al. (2018) [31]	propósito geral	vietnamita	<i>dependency parsing</i>	N/A	<i>dataset</i> de teste	ontologia
Eshghishargh et al. (2018) [32]	neurologia	inglês	palavra-chave	N/A	avaliar intenção	ontologia
Santoso et al. (2018) [33]	Questão universidade	indonésio	<i>DialogFlow</i>	N/A	<i>dataset</i> de teste	ontologia
Abdi et al. (2018) [34]	física	inglês	palavra-chave	N/A	com usuário	ontologia
Wanner et al. (2017) [35]	Coleta informação de saúde	espanhol, alemão, polonês	regras	N/A	usuários	ontologia
Toniuc et al. (2017) [36]	informar mudança climática	inglês	API.AI	N/A	N/A	ontologia
Kincaid et al. (2017) [37]	engenharia	inglês	<i>entity recognition</i>	cache	N/A	ontologia
Garrido et al. (2017) [38]	pontos turísticos	inglês e espanhol	palavra-chave	N/A	com usuários	ontologia
Devi, M. et al. (2017) [39]	agricultura	inglês	pelas entidades e relações	N/A	<i>dataset</i> teste	ontologia
Albarghothi et al. (2017) [40]	patologias	árabe	palavra-chave	N/A	<i>dataset</i> com teste	ontologia
Qin et al. (2020) [25]	Prover informações sanitárias	chinês	a partir das entidade e relações	N/A	N/A	ontologia

- Qin et al. (2020) [25] construíram uma interface de QA com foco em prover informações sanitárias a respeito de produtos alimentícios produzidos na China. Atualmente, só é possível interagir com a ferramenta em chinês. O sistema identifica intenções nas perguntas a partir de entidades e relações extraídas destas por meio de regras e palavras chave, porém a ferramenta não lida com contexto entre as perguntas feitas pelos usuários. Não foi feito qualquer tipo de avaliação da ferramenta por parte dos autores. A ferramenta utiliza uma ontologia para persistir os dados em um banco de triplas e converte a pergunta do usuário em consultas SPARQL;
- Li et al. (2020) [26] desenvolveram um sistema de QA factual a respeito de basquete; não apenas do jogo em si, mas sobre os times, jogadores, estádios etc. O idioma suportado pela ferramenta é apenas o chinês. A intenção das perguntas feitas pelos usuários é capturada a partir do cálculo de similaridade desta com outras questões contidas em um banco de dados previamente construído. Quando essa similaridade não atinge um nível esperado, a intenção é dada a partir das entidades extraídas da pergunta a partir de um modelo de reconhecimento de entidades que os autores desenvolveram. A ferramenta não lida com contexto entre perguntas. Os autores avaliaram apenas uma pequena parte da ferramenta, mais especificamente os modelos de reconhecimento de entidades criados. A ferramenta utiliza um *knowledge graph* para descrição do domínio e persistência os dados, consultando essa base para responder às questões feitas pelos usuários;
- Kulkarni et al. (2019) [27] construíram um sistema de QA chamado ProductQnA, capaz de responder questão a respeito de produtos, como suas especificações, avaliações, garantia etc. em ambientes de *e-commerce*. O sistema proposto identifica a intenção na pergunta a partir da similaridade entre questões e respostas (contidas em um banco previamente construído) utilizando *embeddings*. Os autores utilizam ainda o reconhecimento de entidades nas questões por expressões regulares, *gazetteer* e modelos de *machine learning* (mais especificamente o k-NN) para auxiliar na identificação dessas intenções. A ferramenta criada não lida com contexto entre perguntas. Os autores avaliaram a ferramenta calculando o seu *precision* e *recall* ao responder um conjunto de perguntas e respostas, comparando seu desempenho contra um modelo de *baseline* (mais especificamente um *IDF-vector-average*). Toda essa análise foi feita utilizando um *dataset* de perguntas e respostas que os próprios autores construíram. Os autores se basearam em uma ontologia para criação das

- entidades reconhecidas pelo modelo que utilizaram no reconhecimento de intenção nas perguntas dos usuários;
- Kadariya et al. (2019) [28] construíram um *chatbot* chamado kBot, focado no acompanhamento de pacientes com asma, respondendo perguntas a respeito de sintomas e até mesmo cruzando essas informações com dados climáticos. A ferramenta é capaz de responder apenas perguntas feitas em inglês. Os autores utilizaram a ferramenta *DialogFlow* da *Google* para identificação das intenções, e mantém o contexto da conversa armazenado em um banco *NoSQL*, carregando-o sempre que inicia a conversa com um usuário, porém os autores não entram em muitos detalhes sobre como esse contexto é construído. Os autores realizaram testes de usabilidade com 16 usuários (8 médicos e 8 pesquisadores) simulando cenários de conversas e coletando as impressões de uso por meio de questionários. O conhecimento de domínio a respeito de asma foi extraído manualmente de várias fontes pelos autores e persistido de forma estruturada em um banco *NoSQL*, sendo então utilizados para construção das regras de interação no *DialogFlow*;
 - Avila et al. (2019) [29] desenvolveram uma *chatbot* chamado MediBot, focado em responder perguntas a respeito de medicamentos produzidos no Brasil, fornecendo informações a respeito de contra indicações, riscos etc. O *chatbot* suporta apenas diálogos em português e identifica a intenção das perguntas feitas pelos usuários por meio de palavras-chave. O sistema mantém o contexto dos diálogos armazenados em um banco *NoSQL* (mais especificamente o *MongoDB*) e o carrega sempre que inicia a conversa com um usuário, mas os autores não entram em muitos detalhes sobre como o contexto é construído. A ferramenta foi avaliada com 10 usuários, analisando o tempo e taxa de sucesso destes em responder 6 perguntas de diferentes níveis de dificuldade utilizando o *chatbot*. A ferramenta usa diferentes fontes de dados com estruturas próprias e construiu uma camada de abstração baseada em uma única ontologia para tornar possível a construção de consultas utilizando *SPARQL* na base de conhecimento criada para o *chatbot*;
 - Mittal et al. (2018) [30] apresentaram uma interface conversacional focada na consulta em base de dados relacionais, que converte perguntas feitas em linguagem natural (apenas em inglês) em *queries SQL*. Os autores não deixam claro no texto como identificam as intenções nos diálogos. A ferramenta não mantém o estado do usuário, porém repassa uma representação do estado da conversa para que o usuário repasse-o a cada interação com o servidor. A aplicação utiliza esse estado para lidar

com o contexto, mas os autores não deixam muita claro como isso é feito. A ferramenta foi avaliada de forma manual pelos autores baseados em 200 pares de perguntas e respostas. Aparentemente os autores utilizam alguma forma de incluir uma ontologia para descrever o domínio da base de dados consultada (utilizaram uma ontologia de finanças para exemplificar o uso), porém não entram em muitos detalhes de como funciona de fato essa incorporação da ontologia;

- Le-Hong et al. (2018) [31] desenvolveram um *chatbot* de propósito geral focado em responder perguntas em linguagem natural feitas em vietnamita. O *chatbot* é capaz de responder perguntas como “O que é o Facebook?”, “Quem é o presidente dos EUA?” etc. A ferramenta identifica a intenção no diálogo utilizando uma técnica baseada em aplicar um modelo de *dependency parsing* na pergunta junto com a identificação de entidades e relações por meio de regras e regressores logísticos. A ferramenta não lida com o contexto dos diálogos. Os autores não fizeram teste de usabilidade da ferramenta, apenas calcularam a acurácia desta de forma manual, baseados em um conjunto de 879 pares de perguntas e respostas criado pelos próprios autores. A base de dados que o trabalho utilizou (DBPedia) é baseada em um conjunto de ontologias, e as *queries* feitas nesta base pelo *chatbot* foram realizadas utilizando a linguagem Cypher. Logo, as entidades e relações identificadas nas perguntas feitas pelos usuários tiveram que ser mapeadas para as classes e propriedades das ontologias contidas na base de dados;
- Eshghishargh et al. (2018) [32] construíram um sistema de QA focado em responder perguntas do domínio de saúde, mais especificamente o de neurologia. O sistema suporta apenas perguntas feitas em inglês. A ferramenta identifica a intenção das perguntas feitas pelo usuário a partir de palavras-chave. O sistema não lida com o contexto entre as perguntas. Os autores avaliaram apenas uma parte do sistema, comparando a abordagem do trabalho para identificação da intenção na pergunta baseada em palavras-chave com outros modelos de *machine learning* (*Naïve-Bayes*, *SVM*, *KNN* e *Random Forest*). Toda a consulta à base de conhecimento para obtenção das respostas é feita em SPARQL; tais consultas são construídas a partir do mapeamento entre palavras-chave e entidades descritas nas ontologias utilizadas no trabalho. Os autores se aproveitam ainda do uso de ontologias para aplicar técnicas de expansão das *queries*;
- Santoso et al. (2018) [33] criaram um *chatbot* chamado DINA (*Dinus Intelligent Assistance*), focado em responder questões burocráticas de uma

universidade indonésia, mais especificamente a Universitas Dian Nuswantoro. O *chatbot* suporta apenas perguntas feitas em indonésio. A ferramenta utiliza o DialogFlow da Google para identificar as intenções das perguntas por meio de palavras-chaves. O *chatbot* não lida com contexto. Os autores testaram a ferramenta manualmente, escolhendo 10 amostras aleatórias de um *dataset* contendo pares de perguntas e respostas. Nos testes conduzidos, o *chatbot* conseguiu responder corretamente 8 dessas 10 perguntas selecionadas. Os autores se basearam em uma ontologia para criar as intenções cadastradas por eles no *DialogFlow*;

- Abdi et al. (2018) [34] desenvolveram um sistema de QA chamado QAPD, focado em responder perguntas factuais em inglês a respeito do domínio de Física, mais especificamente sobre eletricidade e eletromagnetismo. Ele é capaz de responder perguntas como “Qual é a unidade da resistência?”, “O que é um transformador?” etc. A intenção nas perguntas é capturada por meio de palavras-chave. Os autores testaram o desempenho da ferramenta com 10 usuários que eram especialistas na área, pedindo para que eles fizessem qualquer pergunta e julgassem se a resposta era satisfatória. Avaliaram também o resultado de outras abordagens comparadas à que desenvolveram no trabalho. Os autores construíram uma ontologia que descreve conceitos físicos e utilizaram essa ontologia para a anotação de entidades e relações nas perguntas feitas pelo usuário. Esse reconhecimento de entidades é feito a partir da análise de similaridade entre termos presentes na ontologia e a pergunta feita pelo usuário;
- Wanner et al. (2017) [35] construíram uma interface conversacional chamada KRISTINA, que funciona como um acompanhante para pessoas idosas, coletando informações a respeito do estado de saúde ou ânimo dos usuários por meio de diálogos. A ferramenta suporta atualmente alemão, polonês e espanhol. A identificação das intenções nas perguntas feitas pelos usuários é feita baseada em regras. A ferramenta lida com contexto, porém os autores não deixam claro como isso é feito. Os autores avaliaram a usabilidade do sistema por meio de um questionário com 5 perguntas a respeito das impressões que os usuários tiveram ao utilizar o sistema, só não deixaram claro quantas pessoas participaram dessa avaliação. A ferramenta utiliza uma ontologia para descrever tanto a base de conhecimento da ferramenta, quanto os atos de fala da interface conversacional;
- Toniuc et al. (2017) [36] apresentam um sistema de QA chamada Climobot, com o intuito de esclarecer o público a respeito de mudanças

climáticas. O único idioma atualmente suportado pela ferramenta é o inglês. Os autores utilizaram a ferramenta API.AI da Google junto com técnicas de *textual entailment*, *hierarchical clustering* e similaridade entre vetores para identificação das intenções nas perguntas. A ferramenta não lida com contexto. Os autores não fizeram qualquer tipo de avaliação da ferramenta. Os autores utilizaram o conhecimento de domínio expressos em uma ontologia para auxiliar na configuração das intenções na ferramenta API.AI, e na geração das *queries* SPARQL para busca na base de dados;

- Kincaid et al. (2017) [37] construíram uma interface conversacional chamada Nicky, que responde a perguntas do domínio de engenharia, mais especificamente o de instrumentos de medição. O sistema proposto consegue responder perguntas textuais e de voz em inglês. As intenções das perguntas são identificadas por meio de um processo de *entity recognition*, porém os autores não deixam claro como isso é feito exatamente. A ferramenta mantém o contexto da conversa em um cache que é apagado sempre que um assunto identificado em um determinado momento não faz referência ao que está no cache. Nenhum tipo de avaliação da ferramenta foi conduzida. Os autores criaram uma ontologia para modelar o domínio de instrumentos de medição e executaram *queries* SPARQL para consultar uma base de conhecimento a partir das perguntas feitas pelo usuário;
- Garrido et al. (2017) [38] apresentam uma interface conversacional personificada em um avatar 3D, que é capaz de responder perguntas em inglês e espanhol a respeito de pontos turísticos. A intenção nas perguntas é identificada por meio de *matching* entre palavras-chave descritas em arquivos AIML. A ferramenta não lida com contexto nas perguntas. Foram realizados testes de usabilidade com alguns usuários para avaliação da ferramenta, porém os autores não deixaram claro quantas pessoas participaram desse experimento. Os autores criaram uma ontologia para modelar o domínio de atrações turísticas e utilizaram esta ontologia para enriquecer as respostas dadas pela interface, como por exemplo sugerir o restaurante que tenha no seu menu um prato com alguma ligação histórica com determinado monumento sobre o qual o usuário tenha perguntado;
- Devi, M. et al. (2017) [39] construíram um sistema de QA chamado ADANS, capaz de responder perguntas factuais em inglês do domínio de agricultura, como “Quais doenças afetam o trigo?”. As intenções nas perguntas são identificadas por meio da extração de entidade e relações

da sentença usando *POS-tagging* e *dependency parsing*. A ferramenta não lida com contexto das perguntas. A avaliação da ferramenta foi feita de forma manual pelos autores, calculando o *precision* e *recall* a partir de 30 questões feitas no sistema. Os autores criaram uma ontologia para o domínio de agricultura e utilizaram ela como base de conhecimento para as respostas dadas pelo sistema, criando consultas SPARQL a partir do mapeamento feito entre as entidades e relações identificadas nas perguntas com classes e propriedades dessa ontologia;

- Albarghothi et al. (2017) [40] desenvolveram um sistema de QA capaz de responder perguntas factuais a respeito de patologias. O único idioma suportado pela ferramenta é o árabe. As intenções nas perguntas são identificadas por meio de palavras-chave. A ferramenta não lida com contexto das perguntas. O autores avaliaram a ferramenta calculando o *precision*, *recall*, *f-measure* a partir de 100 perguntas feitas no sistema, validando manualmente as respostas dadas pelo sistema. Os autores criaram uma ontologia para o domínio de patologias e utilizam esta como a base de conhecimento do sistema, criando consultas SPARQL a partir do mapeamento de entidades contidas na ontologia com palavras chaves contidas nas perguntas feitas pelos usuários.

4 Construção do Agente Conversacional

Neste capítulo descrevemos a abordagem utilizada para realizar a construção do nosso agente conversacional de recomendação baseado em ontologia. Iniciamos o capítulo explicando o módulo de *Interactive Question Answering*, detalhando os processos utilizados na identificação do tipo da pergunta e o processo de recuperação de informações na base de conhecimento. Em seguida abordamos o algoritmo proposto para a recomendação de entidades da ontologia.

4.1 Sistema *Interactive Question Answering*

Um dos principais módulos desenvolvidos é o responsável pelo IQA, onde realizamos o processamento da pergunta do usuário, a consulta da resposta no banco e o envio dela para o Bot, bem como o envio de informações para a árvore de recomendação fazer sua sugestão. O nosso QA é dividido nos seguintes módulos, ilustrados na Figura 4.1:

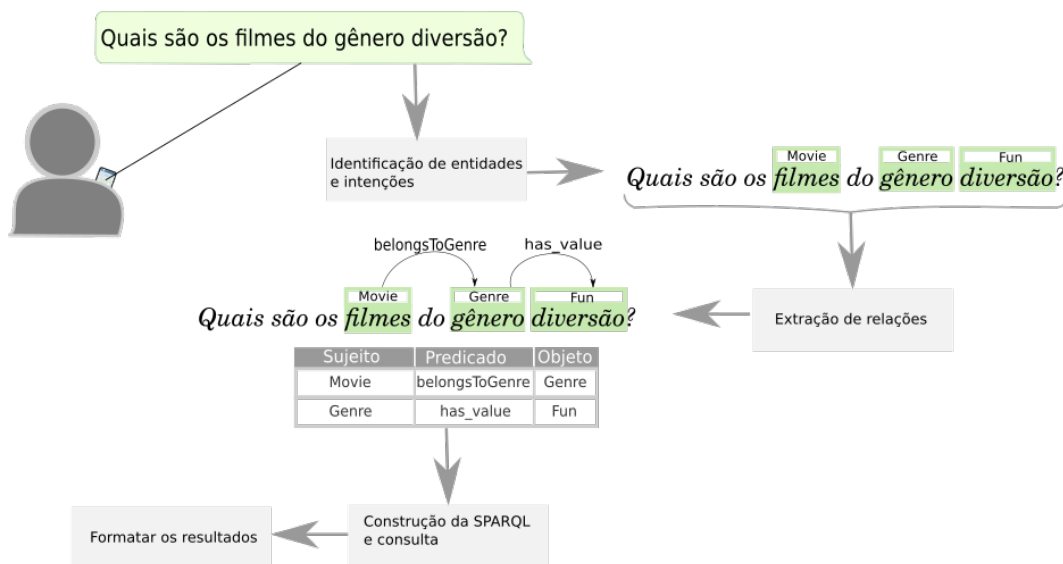


Figura 4.1: Módulos do IQA.

Os módulos do sistema estão destacados como retângulos cinzas, bem como as entradas e saídas de cada um. O primeiro módulo que destacamos é o Identificador de Entidades e Intenções. Ele tem a tarefa de detectar o tipo de

pergunta, se é uma pergunta com múltiplos resultados como “quais os atores de Avatar”, se é uma pergunta de confirmação como “Seria o filme Avatar do gênero Comédia?”, ou se é uma pergunta que necessitamos do contexto para responder como “Atores desse filme”. O segundo módulo se trata do extrator de relações que, dada a pergunta do usuário e as entidades marcadas, extrai as relações entre essas entidades, se houver. As relações são divididas em Sujeito, Predicado e Objeto, onde o Predicado é o nome da relação que o Sujeito tem com o Objeto. Após ter as entidades e relações definidas, elas são alimentadas ao módulo de construção de *SPARQL*, que vai montar a consulta a ser realizada. Os resultados são formatados e processados para serem entregues ao Chatbot que devolverá a resposta para o usuário.

Nas próximas seções detalhamos os processos utilizados por esses módulos.

4.1.1

Processamento de Linguagem Natural

Esta subseção trata dos dois primeiros módulos que lidam com o processamento e entendimento da pergunta do usuário, fornecida em formato textual. Esse texto é transformado em um conjunto de triplas de entidades e relações. Nas subseções a seguir detalhamos o processo de transformação do texto em triplas (Subseção 4.1.1.1, Subseção 4.1.1.2 e Subseção 4.1.1.3), como lidamos com perguntas de contexto (Subseção 4.1.1.4) e a construção da *SPARQL* (Subseção 4.1.1.5) utilizada para recuperar os dados da base de conhecimento.

4.1.1.1

Pré-Processamento

A etapa de pré processamento é encarregada de realizar a “tokenização” do texto, e a retirada de caracteres indesejáveis para as tarefas posteriores. Utilizamos a ferramenta *Rasa* [41] para tal, que faz os seguintes pré-processamentos:

- Tokenização por espaços em branco: O texto é separado em *tokens*, sendo que o caractere de espaço em branco (ou vazio) é a medida de separação. Como exemplo, a frase “Atores do filme Avatar” se transforma nos *tokens*: [“Atores”, “do”, “filme”, “Avatar”].
- Substituição de caracteres especiais: Símbolos especiais como ‘ç’ ou com acentuação como ‘é’, são substituídos respectivamente por ‘c’ e ‘e’.

4.1.1.2

Classificação de Intenções e Extração de Entidades

A tarefa de classificação de Intenção tem como objetivo dizer qual o tipo de pergunta, dentre os seguintes tipos pré-definidos:

- **Select:** Esta intenção abrange perguntas como: “*Quais os atores de Avatar*” ou “*Em que filmes Angelina Jolie atuou?*”
- **Ask:** Esta intenção abrange perguntas que verificam fatos, como “*Avatar é do gênero de Ação?*” ou “*Angelina Jolie é uma Atriz?*”
- **Context:** Esta intenção indica se a pergunta realizada é incompleta, ou seja, se é necessário que sejam inferidas informações do contexto para respondê-la. Como exemplo: “*Quais as atrizes deste filme?*”, onde o filme havia sido referenciado anteriormente.

A Extração de entidades trata de identificar na pergunta as entidades nomeadas de interesse. Na imagem abaixo, ilustramos um exemplo onde as entidades identificadas estão destacadas em verde

Quais são os filmes do gênero diversão?

Figura 4.2: Reconhecimento de entidades nomeadas.

Utilizamos o *framework Rasa* para classificação e extração conjunta de intenções e entidades, cuja arquitetura é ilustrada em alto nível na Figura 4.3.¹

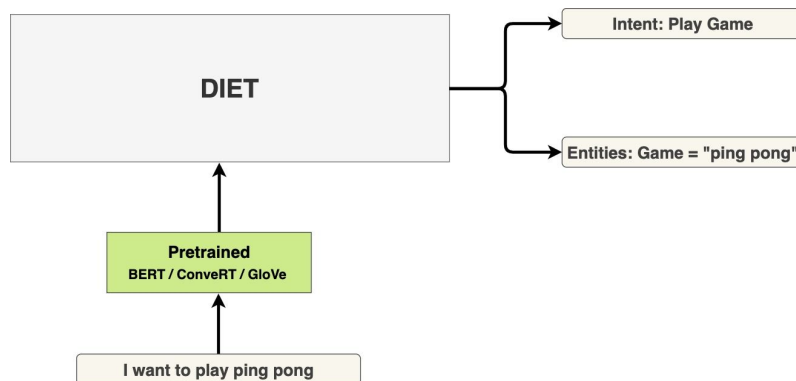


Figura 4.3: Arquitetura DIET.

A pergunta do usuário passa pelo pré-processamento e é alimentada ao modelo de rede neural *transformer* pré-treinado, que pode ser opcionalmente o **BERT**, **ConveRT** ou **GloVe** [42]. A saída será um *embedding*, que é uma

¹<https://blog.rasa.com/introducing-dual-intent-and-entity-transformer-diet-state-of-the-art-performance-on-a-lightweight-architecture/>

representação em forma de vetor de valores reais da questão, onde sentenças similares terão representações vetoriais mais próximas.

O Modelo *DIET* é uma arquitetura particular do *Rasa*, conhecida por ser de fácil e rápido treinamento, afirmando ser 6 vezes mais rápida que os modelos estado-da-arte devido ao uso extensivo de *dropout* nas redes neurais [43].

Para treinamento do modelo, utilizamos uma linguagem específica de Domínio chamada *Chatette*,² que é utilizada para geração de *datasets* para NLP. Nela, construímos *templates* com as intenções e entidades marcadas, bem como definimos partes das sentenças que serão utilizadas no treinamento. Na Figura 4.4 mostramos o arquivo de *templates*: as intenções são definidas como %[nome_intenção]; as entidades são marcadas como @[nome_entidade]; e a quantidade de exemplos gerados para uma dada intenção é definida ao lado da sua declaração. Ainda no *Chatette* é possível permitir uma maior variedade do *dataset* ao usar outras funções que são precedidas pelo símbolo ~. Essas funções serão trocadas pelos textos definidos em seu escopo, no exemplo a seguir, a função ~ [ola] poderá ser substituída por *oi* ou *bom dia*, por exemplo.

```
%[filtra_por_genero](5)
  ~[ola?] ~[poderia dizer?pode] ~[diga me?!pode] filmes ~[do genero] @[gênero] [por favor?]?

~[ola]
  oi
  olá
  bom dia
  boa tarde
  boa noite

~[diga me]
  [me?pronome] [diga|conte|mostre|liste] [me?!pronome]

~[poderia dizer]
  [pode|poderia] me [dizer|listar|mostrar|contar]

~[do genero]
  do gênero
  que são do gênero
  que pertencem ao gênero
  que são classificados no gênero

@[gênero]
  ação
  drama
  comédia
  infantil
```

Figura 4.4: Exemplo de template utilizado no *Chatette*.

Abaixo mostramos algumas possíveis sentenças geradas para o *template* definido na Figura 4.4.

- boa noite poderia me mostrar filmes que pertencem ao gênero infantil por favor?

²<https://github.com/SimGus/Chatette/wiki>

- bom dia mostre me filmes que são do gênero drama por favor?
- oi mostre me filmes que pertencem ao gênero infantil por favor?
- poderia me dizer filmes que pertencem ao gênero ação?
- poderia me listar filmes que são do gênero ação por favor?

Por fim, construímos um conjunto de 90 *templates* que abrangem as 3 intenções, com 3000 exemplos cada, separados igualmente, permitindo o balanceamento do *dataset*. Dentre os exemplos gerados, dividimos 80% para treino e 20% para teste para cada intenção.

Treinamos com o classificador de intenções e entidades do *Rasa*, que permite personalizar de acordo com as preferências do usuário. Utilizamos então modelos pré-treinados em português, para facilitar o aprendizado e diminuir o tempo necessário de treinamento.

Obtivemos bons resultados no treino e validação, com o *Rasa* relatando 95% da métrica *f1* na detecção de intenções, e 93% na detecção de entidades. É importante frisar, no entanto, que esse é um *dataset* ainda pequeno, contendo em si apenas *templates* com que imaginamos contemplar algumas das perguntas mais comuns do usuário, portanto não podendo servir de métrica para perguntas fora do escopo limitado.

4.1.1.3 Extração de Relação

A tarefa de extração de relação consiste em fazer uma ligação entre entidades que possuem um relacionamento, e atribuir um nome a esse relacionamento. A Figura 4.5 destaca os relacionamentos marcados para as entidades identificadas. Nela, fazemos uma ligação entre as entidades *Movie* e *Genre*, de nome *BelongsToGenre*, e de *Genre* com *Fun*, de nome *has_value*.

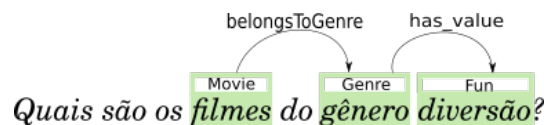


Figura 4.5: Reconhecimento de relações.

Existem muitas abordagens para esta tarefa, e com base no sucesso recente de redes neurais na área de NLP, adotamos a estratégia de classificação de texto de modo supervisionado. A estratégia é dividida nas seguintes etapas:

- Anotamos as regras que constroem o *dataset* de Entidades, com as relações entre as entidades. Consideramos essas anotações como positivos, e todas as relações inexistentes como negativos. Como exemplo, em “Quais os filmes do gênero diversão?”:

- (*Movie*, *Genre*) e (*Genre*, *Fun*) são pares positivos
 - (*Genre*, *Movie*), (*Fun*, *Genre*), (*Movie*, *Fun*), (*Fun*, *Movie*) são pares negativos
- Alimentamos o classificador com a pergunta mais o par da amostra em questão. Ele vai ser treinado para inferir o *label* (positivo ou negativo) com base nessas informações. Para cada sentença anotada com suas entidades, fazemos a permutação com agrupamento 2 a 2 das entidades. Por exemplo:
- (“Quais os filmes do gênero diversão?”, (*Movie*, *Genre*)), Positivo
 - (“Quais os filmes do gênero diversão?”, (*Genre*, *Movie*)), Negativo
 - (“Quais os filmes do gênero diversão?”, (*Movie*, *Fun*)), Positivo
 - (“Quais os filmes do gênero diversão?”, (*Fun*, *Movie*)), Negativo
 - (“Quais os filmes do gênero diversão?”, (*Genre*, *Fun*)), Positivo
 - (“Quais os filmes do gênero diversão?”, (*Fun*, *Genre*)), Negativo
- Como classificador utilizamos um *transformer*, já pré-treinado em corpus de linguagem portuguesa, facilitando assim o aprendizado nesta tarefa.
- Na inferência, simplesmente passamos ao classificador todas as possíveis relações entre as entidades identificadas, para que ele as classifique como positivas ou negativas. Utilizamos então as relações positivas para as próximas etapas, lembrando que, como o número de entidades é geralmente pequeno, a inferência em todas as permutações vai ser rápida.
- Nomeamos a relação entre os pares classificados como positivo por meio de conhecimento de domínio da ontologia.

Geramos um *dataset* com a mesma quantidade de exemplos do gerado para a identificação de entidades e intenções, seguindo o mesmo critério de particionamento de treino e teste, e acrescentamos as relações nesses *templates*. Utilizamos um modelo *transformer BERT* [44] pré-treinado em dados de múltiplos idiomas. Realizamos *fine tuning* para classificação binária, obtendo uma acurácia de 94%.

4.1.1.4

Contexto

Perguntas de contexto são aquelas que não fornecem informações suficiente para respondê-las, necessitando recorrer a informações extraídas de mensagens anteriores. Neste trabalho realizamos o seguinte tipo de tratamento:

- Referência a entidades: Nesse caso, são perguntas sobre entidades já mencionadas. Dado que a primeira pergunta foi “Quais os atores de Avatar”, podemos obter mais informações sobre esse filme através de perguntas como “e suas atrizes?” ou “qual foi o orçamento desse filme?”

Para tratar essas questões, precisamos primeiro identificá-las como questões que precisam de contexto. Na etapa de Classificação de Intenção e Extração de entidades, criamos no *dataset* exemplos dessas questões e as anotamos com a intenção de contexto.

Para a referência a entidades, verificamos se a entidade requerida, como atrizes em “e suas atrizes?”, é uma propriedade de alguma das entidades já mencionadas. Ao percorrer o histórico, vemos que “atrizes” é uma propriedade da entidade “*Movie*”, que foi mencionada na primeira pergunta “Quais os atores de Avatar?”. A busca o histórico vai da mais recente às mais antigas, e, caso não tivesse encontrado nenhuma entidade da qual atrizes são uma propriedade, o *IQA* retorna resultado não encontrado.

Enquanto a forma de inferência de contexto pode ser vista como simplista, ela permite fracionar até certo ponto as questões que podem ser feitas, realçando a características de se utilizar *IQA* e *chatbots*.

4.1.1.5

Construtor de SPARQLs

Nesta seção descrevemos o módulo responsável pela construção de *SPARQL* a partir da intenção e do conjunto de triplas extraídas na etapa de extração de relações. Inicialmente definimos um *template* que permite o mapeamento das triplas para o trecho *SPARQL* correspondente. Como ilustrado na Figura 4.6, dividimos o *template* em quatro tipos, sendo eles: classe – define o tipo da variável (sua classe); propriedade – representa relação entre duas variáveis ligadas por um *object property* ou *data property*; *form* de consulta – que define se a consulta é do tipo *SELECT* ou *ASK*; e outros – a primeira instrução atribui valores a uma dada variável e a segunda instrução indica que será contada a quantidade de resultados.

Na Figura 4.7 demonstramos o uso dos *templates* na construção da *SPARQL*. Para a primeira tripla definimos o tipo de cada variável através

Templates			
Class Template		Property Template	
Award	var_name a :Award .	belongsToGenre	var_name :belongsToGenre var_name_2.
Person	var_name a dbo:Person .	hasActress	var_name :hasActress var_name_2.
Actor	var_name a dbo:Actor .	hasMaleActor	var_name :hasMaleActor var_name_2.
Movie	var_name a :Movie .	title	var_name :title var_name_2.
TVSeries	var_name a :TVSeries .	birthName	var_name :birthName var_name_2.
Others		Query Form Template	
VALUES var_name {uri_list}		SELECT vars_list WHERE {body}	
COUNT (*)		ASK WHERE {body}	

Figura 4.6: *Template* utilizado para a construção das SPARQLs.

do *Class Template*, a variável do tipo *movie* será mapeada como *?movie* a *:Movie* e *genre* como *?genre* a *:Genre*. Em seguida verificamos a propriedade de ligação, nesse exemplo *belongsToGenre* irá gerar a sub-SPARQL *?movie :belongsToGenre ?genre*. Enquanto construímos o corpo da *SPARQL*, guardamos as possíveis variáveis de interesse considerando aquelas que não possuem um valor associado. Variáveis acompanhadas da ligação *has_value* são desconsideradas como variáveis de interesse da consulta.

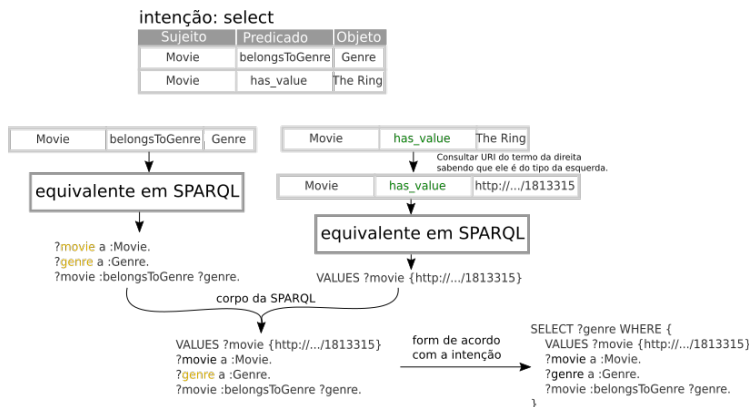


Figura 4.7: Processo de construção das SPARQLs.

4.2 Sistema de Recomendações por Ontologia

A seguir descrevemos o funcionamento do sistema de recomendação baseado na ontologia. Inicialmente precisamos converter o arquivo da ontologia no objeto *OntologyTrees*. Esse objeto contém três árvores, onde cada árvore representa respectivamente a hierarquia de classes, propriedades de objetos e propriedades de dados (Figura 4.8). Cada nó possui referência ao seu pai e aos seus filhos, *label*, uma variável de peso, referência aos seus *domains*³ e *ranges*⁴.

³<https://www.w3.org/TR/owl-ref/#domain-def>

⁴<https://www.w3.org/TR/owl-ref/#range-def>

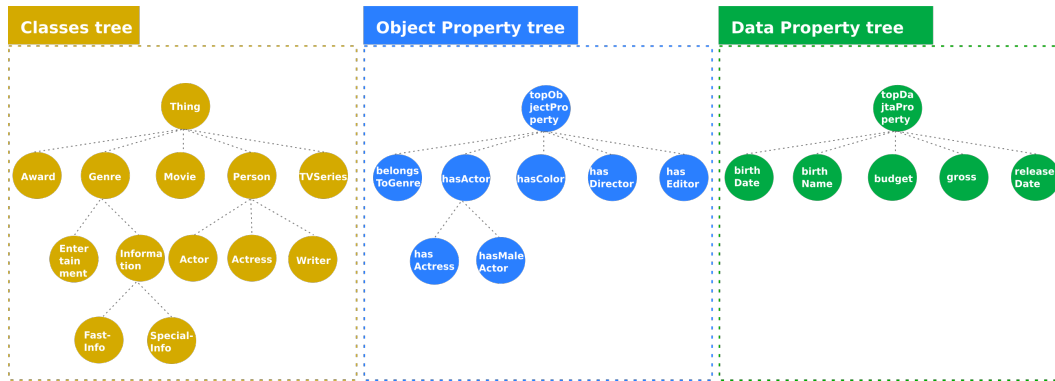


Figura 4.8: Árvores representando as hierarquias na ontologia.

4.2.1 Busca

Para coletar os nós de interesse, percorremos as árvores da esquerda para direita, um nível de cada vez, essa varredura também conhecida como busca em largura. Entretanto, elementos pouco relacionados podem ser retornados nesta visitação, e para lidar com isso definimos alguns critérios restritivos, tais como: a quantidade de sugestões; ordem de exibição; posição do nó ancestral e profundidade da consulta; limiar com base no *domain* e/ou *range*.

Quando limitamos um nó com base no *domain* ou *range* do nó de referência, podemos ter uma maior garantia de que aquela propriedade ou classe está sendo substituída por um elemento que mantém a estrutura de coerência da frase. Por exemplo, se no mesmo nível da árvore possuímos as propriedades **atuar** e **pertence ao gênero**, **atuar** tem o *domain* de **Pessoa** diferente de **pertence ao gênero** que será do tipo *Movie* ou *Serie*. Portanto, se aplicarmos a restrição, elas nunca poderão ser substituídas entre si, pois ambas possuem *domain* e *range* distintos. A substituição é realizada com base no nó que representa o atributo o qual desejamos substituir na frase, chamado de nó de referência. Já o nó ancestral é utilizado como raiz da nossa busca, sendo que é um ancestral do nó de referência, e sua posição parental (*parent*, *grandparent* etc.) é determinada pelo *family_position*.

O *family_position* corresponde ao nível em relação ao nó de referência, podendo assumir valores menores ou iguais a zero. Por questões de explicação, na Figura 4.9 demonstramos a recomendação baseada em apenas na **classe**, consideramos o elemento diversão que marcamos com o rótulo *Fun*. Destacamos a posição de cada ancestral em roxo na Figura 4.9.

Mostramos na Figura 4.10 os nós que serão sugeridos no lugar do nó de referência, de acordo as configurações definidas no quadro cinza da imagem correspondente.

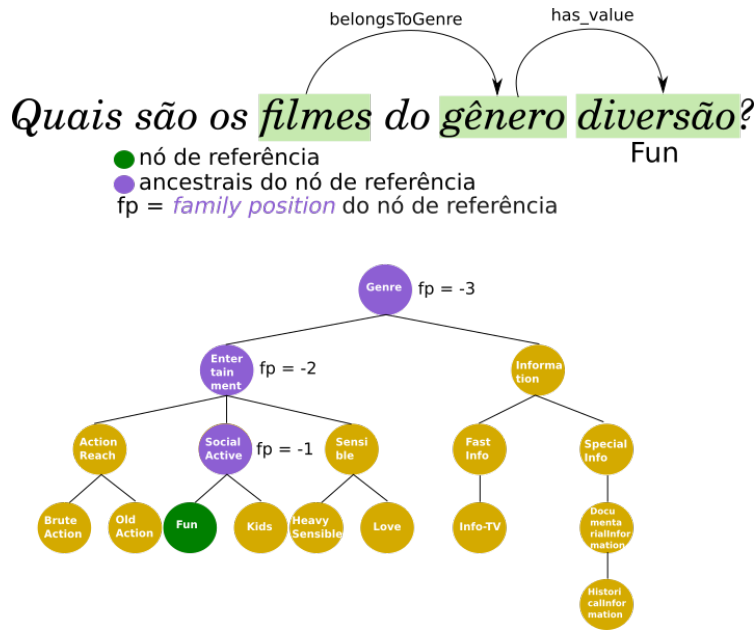


Figura 4.9: Árvores de gêneros.

4.2.2 Embeddings

A última estratégia se trata de ordenar as recomendações por meio do uso de word *embeddings*. Nesse caso, ordenamos pela similaridade de *coseno* as entidades retornadas pela busca da árvore. Para esta abordagem, utilizamos os *embeddings wang2vec* com uso do *skip-gram* [1], escolhidos através de uma simples verificação de ordenação das recomendações entre vários modelos de *embeddings*. Como exemplo da ordenação, mostramos uma comparação entre uma ordenação aleatória da pergunta ‘filmes de gênero infantil’, e outra com ordenação por similaridade com a palavra ‘infantil’

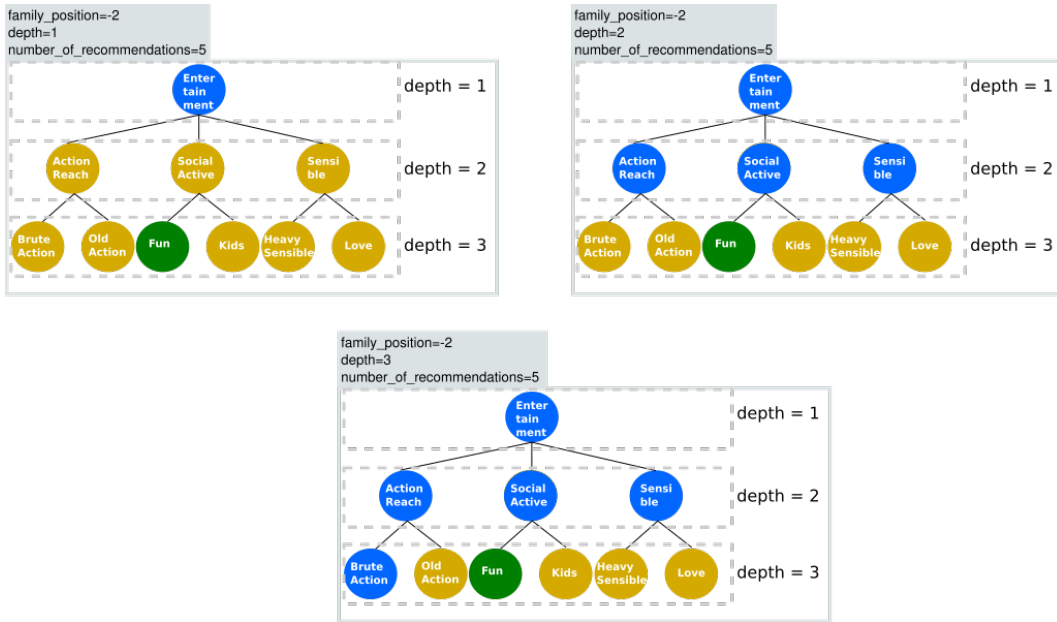


Figura 4.10: Árvores de recomendação.

PUC-Rio - Certificação Digital Nº 1813315/CA

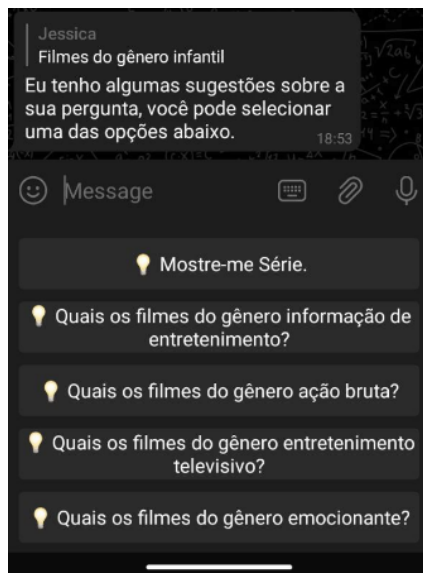


Figura 4.11: Ordenação aleatória.

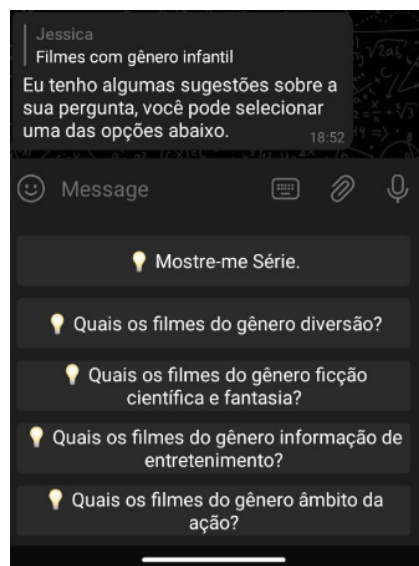


Figura 4.12: Ordenação por similaridade dos *embeddings*.

5 Arquitetura

O nosso sistema conversacional foi desenvolvido para o aplicativo de mensagens instantâneas *Telegram*¹ devido à facilidade de uso da API de *chatbots*. Funcionalidades específicas dessa plataforma foram essenciais na usabilidade do sistema, tais como os recursos de edição de mensagens e a associação de botões às mensagens que foram utilizadas no processo de paginação dos resultados que detalhamos na Seção 5.2.

Organizamos essa arquitetura em quatro grandes módulos: (i) *Models*, que lida com recuperação e tratamento de dados vindos do banco de dados e do IQA; (ii) *Dialogs*, que cuida da formatação da mensagem e como ela será exibida ao usuário; (iii) *Controllers*, que captura as ações do usuário e as redireciona aos componentes *Dialogs* e *Models*; e (iv) *Templates*, que contém as configurações de como serão exibidas as mensagens pelo módulo *Dialogs*. A visão geral do sistema aparece ilustrada na Figura 5.1. Nas próximas seções detalhamos cada um desses blocos.

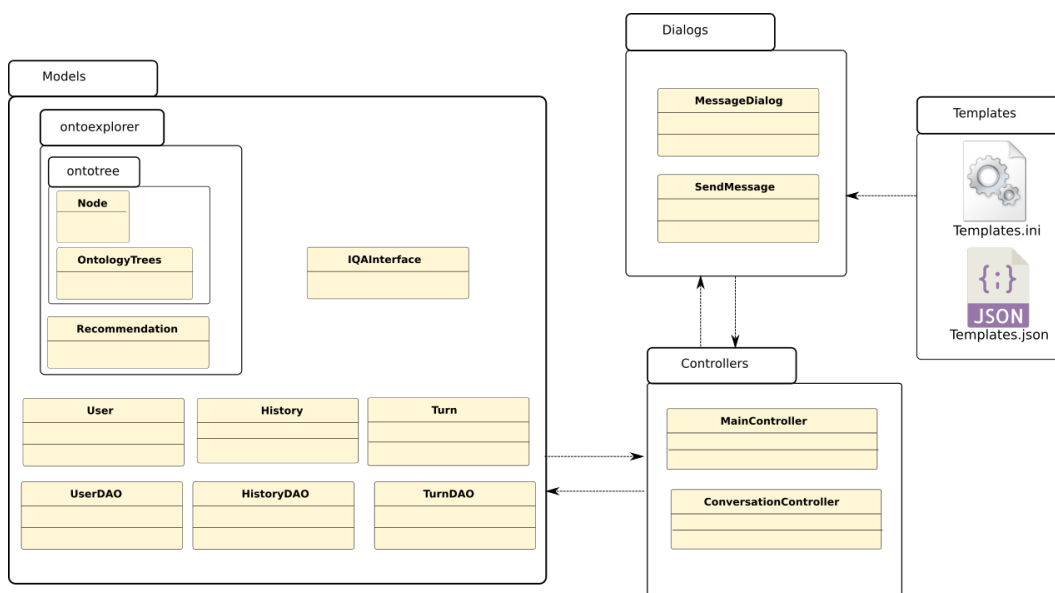


Figura 5.1: Visão geral dos componentes do *chatbot*.

¹<https://telegram.org/>

5.1 Templates

Os arquivos responsáveis pelos ajustes na aparência e exibição de textos-padrão estão definidos nesta pasta. A partir das configurações do arquivo *template.ini* (Código 1), podemos manipular a forma como os resultados serão exibidos ao usuário, como a quantidade de respostas exibidas por vez (*page_size*), posição dos nomes da coluna (*col_name_pos*), separador de linha e coluna das respostas.

```
[answer]
page_size=5
col_name_pos=top
; Separador pode conter emojis, sinais de pontuações, texto
; \n para quebra de linha
; \t para tabulação
col_sep=\t
line_sep=\n
number_format=Resultado {}: \n
```

Listing 1: Arquivo de ajuste visual das mensagens de resposta.

As mensagens-padrão exibidas na interface conversacional são definidas no arquivo *template.json*. No Código 2 mostramos o *welcome*, que se refere à mensagem inicial exibida em nosso sistema e o *done*, que representa a mensagem exibida após o fim de um fluxo de conversa. O campo *src_type* serve para indicar o tipo do endereço da mídia informada no *src*, que pode ser *photo*, *animated* ou *video*. Esses campos estão presentes apenas nas mensagens de **boas-vindas**, **sobre** e **ajuda**.

5.2 Dialogs

Este módulo contém duas classes, sendo: (i) *MessageDialog*, onde carregamos as mensagens-padrão na variável *template* e as configurações de exibição em *config*. As mensagens de boas-vindas, sobre e ajuda são exibidas de forma paginada, onde cada posição do vetor definido no *JSON* (Código 2) é exibida de cada vez.

As respostas enviadas pelo nosso *bot* são processadas pelos métodos: (i) *reply_answer()*, que lida com a formatação da mensagem recebida e (ii) *edit_answer()* que substitui a mensagem antiga pela próxima mensagem sempre que o usuário utilizar o comando de paginação. Por fim, as mensagens são enviadas ao cliente através dos métodos da classe *SendMessage*.

```

{
  "welcome": [{
    "text": "Bem vindo(a) ao *IQA IMDB (bot)*!\n\nEste
    ↪ bot tem como objetivo responder perguntas no
    ↪ domínio do IMBD.\n\n    Dicas e instruções de
    ↪ uso podem ser acessadas através do comando
    ↪ /help.\n    Informações sobre o bot estão
    ↪ disponíveis no comando /about.\n\n*Aperte em um
    ↪ dos botões abaixo para ver mais informações.*",
    ↪ "src": "", "src_type": ""
  },{
    "text": "*Como utilizar este bot:* \n\n 1- Use o
    ↪ comando `/start_iqa` para iniciar a conversa.\n
    ↪ Você poderá usar esse comando novamente após
    ↪ finalizar uma conversa com o comando `/done`.\n
    ↪ 2- _Envie a pergunta desejada no campo de enviar
    ↪ mensagem._\n    Você poderá ver alguns exemplos
    ↪ de perguntas ao utilizar o comando
    ↪ `/examples`.\n 3- Para finalizar a conversa
    ↪ utilize o comando `/done`.\n\n*Para ver mais
    ↪ dicas e instruções de uso, utilize o comando
    ↪ /help.*", "src": "", "src_type": ""
  }
  ],
  "done": {
    "redirect": {
      "text": "Conversa finalizada, até a próxima. Você
      ↪ pode iniciar outra conversa utilizando o comando
      ↪ `/start_iqa`. Para avaliar este bot acesse o
      ↪ [questionário](https://link-do-form) e informe o
      ↪ seguinte ID `{0}` no campo requisitado."
    },
    "default": {
      "text": "Você deve enviar pelo menos uma mensagem
      ↪ antes de finalizar a conversa."
    }
  }
}

```

Listing 2: Arquivo contendo as mensagens padrões exibidas pelo bot.

A resposta retornada pelo IQA pode ser uma lista contendo múltiplos resultados. A fim de lidar com esse problema, decidimos realizar um tipo de paginação sobre essa lista. Para isso utilizamos as configurações de paginação definidas no arquivo *templates.ini*: para uma lista com N resultados, dividimos essa quantidade pelo *page_size* e assim obtemos a quantidade de páginas. Na Figura 5.3 mostramos a formatação dos resultados considerando os parâmetros

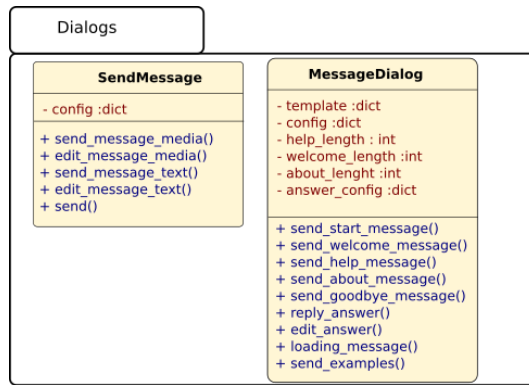


Figura 5.2: Detalhes das classes do módulo *Dialogs*.

definidos no Código 1.

No exemplo, a posição das colunas *Ator* e *Data de nascimento* ficou no topo devido ao valor do parâmetro *col_name_pos*, que pode assumir os valores *top*, *bottom* e *default*; este último indica que as colunas serão exibidas ao lado de cada resultado. O separador de coluna (*col_sep*) indica como as colunas devem ser exibidas. No exemplo, estão lado a lado, o separador de linha é similar ao de coluna, só que indica separação entre cada linha do resultado. Por fim, temos a formatação da numeração exibida para cada resultado, que varia de 1 a N.



Figura 5.3: Mensagem formatada para paginação de acordo com *templates.ini*

5.3 Models

Um usuário pode possuir um ou mais históricos de conversa. Cada histórico é composto por um conjunto de turnos, que são definidos como pares de mensagem do usuário e a resposta obtida pelo IQA. As informações coletadas na conversa são salvas em um banco de dados, no nosso caso

PostgreSQL, estando disponíveis para o desenvolvedor recuperá-las quando julgar necessário. Neste módulo (Figura 5.4), realizamos estas operações de consulta e escrita no banco de dados, além de consultas ao IQA, que são feitas através de requisições HTTP, seguindo o formato `http://localhost:5000/iqa-imdb/perform_search?q=<pergunta>`.

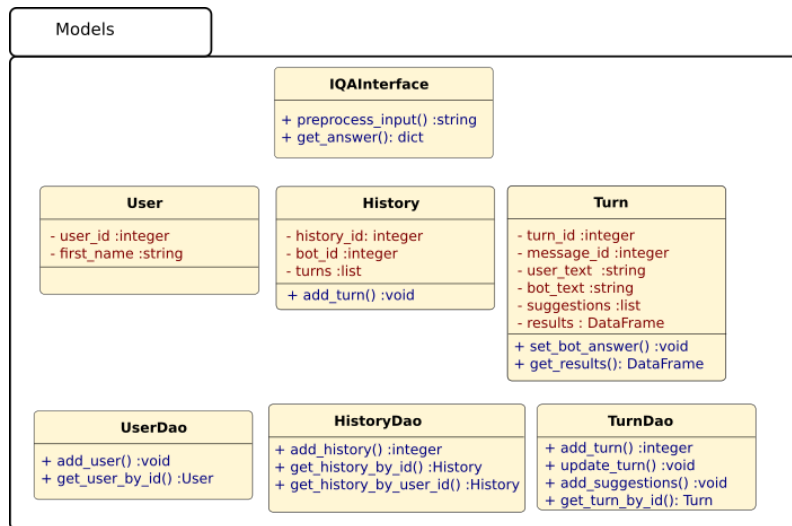


Figura 5.4: Detalhes das classes do módulo *Models*.

Na primeira vez que o usuário inicia a conversa, registramos o seu id e nome de usuário. Sempre que o usuário inicia uma novo fluxo de conversa que utilize nosso IQA, criamos um novo histórico, que irá coletar todas as mensagens da conversa enquanto não houver a finalização desse fluxo. O objeto do tipo *Turn* irá conter a data em que ele for registrado, texto do usuário, sugestões relacionadas e resposta enviada pelo *bot*.

5.3.1 Comunicação com IQA e recomendações

A resposta obtida pelo IQA é apresentada como um JSON, contendo os seguintes campos:

- **text**: Texto em linguagem natural representando a resposta da pergunta ou descrição dos resultados.
- **results**: Dicionário representando os resultados encontrados, onde as chaves indicam o que foi retornado, conforme ilustramos no exemplo a seguir:
- **relations**: As relações extraídas conforme descrito na Subseção 4.1.1.3 são reaproveitadas pelo submódulo de recomendações que nomeamos como *ontoexplorer* e são retornadas pelo IQA como uma lista de tuplas.

```

- {
  "Ator": [
    "Nome A", "Nome B", "Nome C", "Nome D", "Nome E",
    ↪ "Nome F", "Nome G", "Nome H", "Nome I", "Nome
    ↪ J", "Nome K"
  ],
  "Data de nascimento": [
    "Data 1", "Data 2", "Data 3", "Data 4", "Data 5",
    ↪ "Data 6", "Data 7", "Data 8", "Data 9", "Data
    ↪ 10", "Data 11"
  ],
}

```

Listing 3: Exemplo de resultados retornados.

- **related**: Lista contendo as perguntas relacionadas, caso o IQA acoplado forneça recomendações; caso contrário, é esperada uma lista vazia.
- **suggestion_text**: Texto explicativo sobre a lista de recomendações exibida; caso não haja, espera-se uma *string* vazia.

Neste módulo também tratamos as operações de extração de sugestões com base na ontologia. Utilizamos os valores do campo *relations* retornados pela consulta ao IQA para invocar a classe de recomendação localizada no submódulo *ontoexplorer*, que representamos na Figura 5.5. A partir de um arquivo da ontologia, construímos o objeto *ontologytrees* e a classe *Recommendation* trabalha sobre esse objeto de acordo com os parâmetros do arquivo de configuração *recommendation.ini* (Código 4). Esse processo é descrito na Subseção 4.2. O parâmetro *order* define a forma como os resultados da visita na árvore serão apresentados, podendo ser fora de ordem específica (*random*) ou aplicando a similaridade semântica (*semantic*) de acordo com *embedding* definido na variável *model_path* e o limiar de similaridade mínimo (*similarity_threshold*).

5.4 Controllers

O usuário interage com o sistema conversacional através do *Telegram* por meio de mensagem de texto, comandos e ações realizadas através de botões. Essas operações são tratadas pelas classes *MainController* e *ConversationController*, que invocam os métodos dos demais módulos de acordo com a ação executada.

As mensagens recebidas são processadas pelo *msg_handler()*, que se comunica com a *IQAIInterface* para obter a resposta. Em seguida, ela é

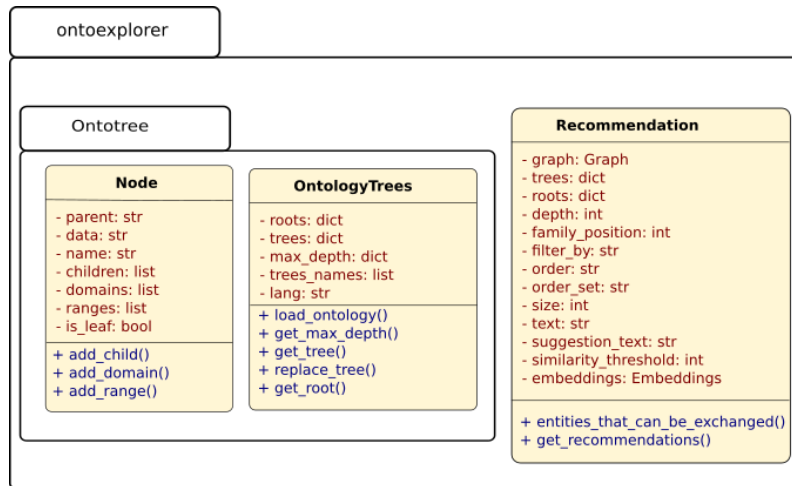


Figura 5.5: Detalhes das classes do submódulo *Ontoexplorer*.

[DEFAULT]

```

; Arquivo contendo a ontologia
ontology_filename=my_ontology.ttl
ontology_format=ttl
rec_text= !! Mostre-me {}.
suggestion_text=Você pode selecionar uma das seguintes sugestões.

; caminho contendo o embedding a ser utilizado.
model_path=embedding.txt
similarity_threshold=-1

; Parâmetros de visita da árvore
family_position=-2
filter_by=both
depth=3
order=
order_set=all
number_of_recommendations=50

```

Listing 4: Parâmetros utilizados pela recomendação.

formatada e enviada através do *Dialogs*. Caso a resposta contenha múltiplas linhas, é realizada a exibição da mensagem de forma parcial, contendo botões em seu rodapé indicando a quantidade de páginas e navegação para esquerda e direita. Quando o usuário clica em um desses botões, o *Controller* chama a operação do *Dialogs* responsável por substituir a mensagem atual pela página correspondente.

Nossa aplicação permite o uso de cinco comandos: */start*, */about*, */help*, */start_iqua* e */done*. O fluxo de interação com o *IQA* acontece entre os comandos */start_iqua* e */done*, conforme ilustramos na Figura 5.7, as mensagens enviada

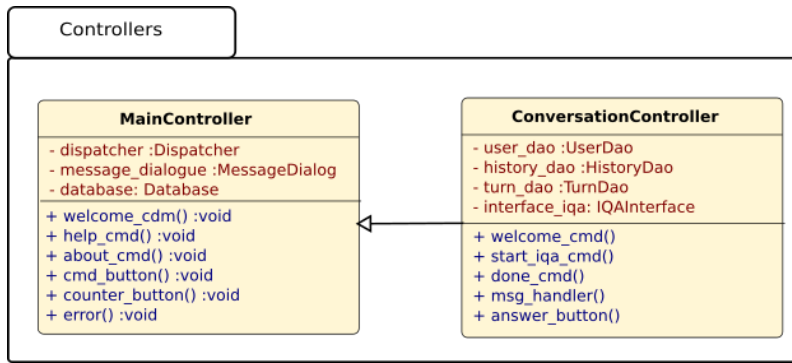


Figura 5.6: Detalhes das classes do módulo *Controllers*.

pelos usuários do tipo texto são gravadas no banco de dados até o final do fluxo.

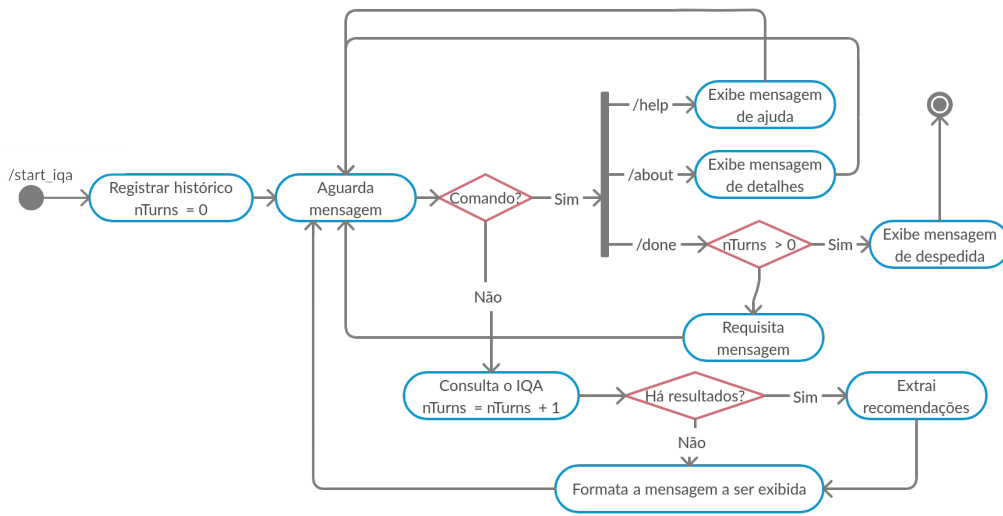


Figura 5.7: Fluxo da conversa com IQA

6 Avaliação

Neste capítulo realizamos uma avaliação comparativa entre os *chatbots* com recomendações em relação ao sem recomendações. Como descrito na Subseção 4.2, a nossa metodologia exige que as propriedades da ontologia possuam *domains* e *ranges*, e para esse estudo utilizamos a base do *IMDb Movie Ontology* que foi desenvolvida por Calvanese et al. (2017) [17].

As recomendações utilizadas em nosso *chatbot* são retornadas pela busca em largura na árvore da ontologia, de acordo com os *domains* e *ranges* de cada propriedade, ou seja, as entidades sugeridas geralmente possuem uma dessas informações em comum. A ordem de exibição das recomendações pode seguir então três estratégias: (i) ordem de visita da árvore; (ii) ordenação aleatória (iii) ordenação por *embeddings*.

Realizamos um estudo com 31 participantes voluntários. Cada participante testou uma versão do *bot* sem recomendação e uma outra, utilizando uma das três estratégias de ordenação. Nesse estudo, dividimos os participantes em seis grupos (Tabela 6.1), onde cada grupo representava a permutação entre os *bots* de cada par. Também desenvolvemos dois roteiros que deveriam ser executados para cada um dos pares. Essas configurações foram elaboradas com o propósito de evitar que os participantes fossem enviesados devido a uma ordem particular dos *bots*.

Grupos			
Grupo	Primeiro Bot	Segundo Bot	Número de Voluntários
1	Sem recomendação	Ordem aleatória	6
2		Ordem de visita na árvore	5
3		Ordenação por <i>Embeddings</i>	5
4	Ordem aleatória	Sem recomendação	5
5	Ordem de visita na árvore		5
6	Ordenação por <i>Embeddings</i>		5

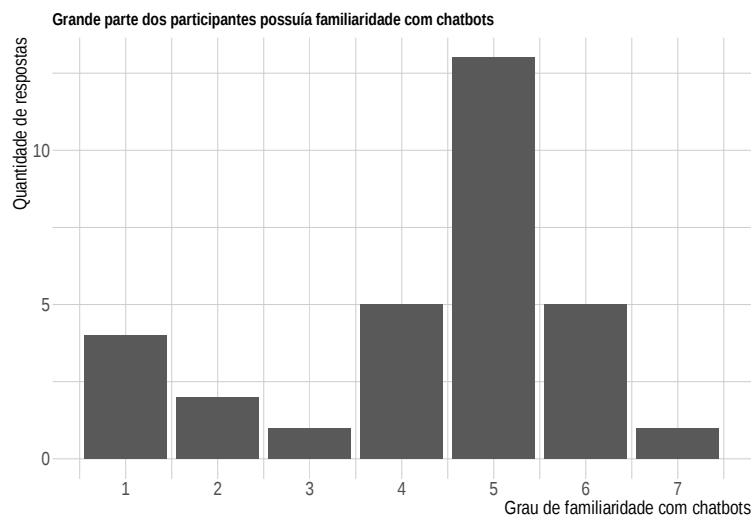
Tabela 6.1: Grupos do experimento

Para avaliar o impacto das recomendações, os voluntários dessa pesquisa responderam dois questionários (Anexo A), que visavam a comparar as abordagens com e sem recomendação. Com os dados coletados da conversa, também medimos a frequência do uso das recomendações. Nas próximas seções anali-

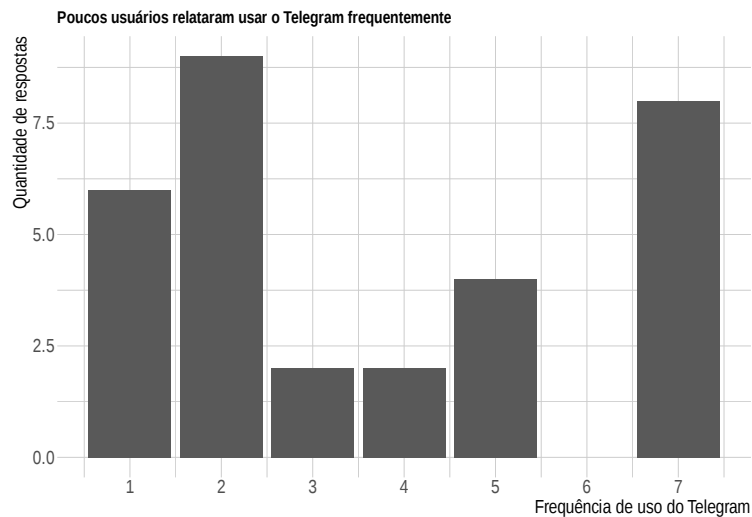
samos sobre as informações coletadas via formulário e histórico da conversa, fazendo um paralelo entre essas abordagens.

6.1 Estatísticas dos participantes

O nosso estudo contou com a participação de 31 voluntários, com idades entre 17 e 54 anos. A maioria dos participantes possuía bastante familiaridade com *chatbots* (Figura 6.1a), indicada por eles numa escala de 1 (nenhuma familiaridade) a 7 (muita familiaridade). Entretanto, grande parcela utilizava o *Telegram* com baixa ou nenhuma frequência (Figura 6.1b).



(a) Familiaridade com bots



(b) Frequência de uso do Telegram

Para avaliação dos bots, listamos aqui afirmações apresentadas aos participantes que consideramos mais relevantes:

- Aprender a utilizar o bot foi fácil para mim.
- O bot foi fácil de usar.
- O bot foi rápido de usar.
- O bot me permitiu fazer boas consultas.
- Eu achei fácil obter do bot o que eu queria.
- Interagir com o bot requereu muito esforço mental.
- Interagir com o bot foi frustrante.
- Eu me sinto confiante em buscar informações no bot.

Para cada afirmação, o usuário deveria indicar sua concordância, variando de discordância total (1) a concordância total (7).

6.2

Estatísticas de tempo

Consideramos os quatro bots diferentes em nossa análise:

- Bot 1: Bot sem recomendação
- Bot 2: Bot com recomendação aleatória
- Bot 3: Bot com recomendação de árvore
- Bot 4: Bot com recomendação de *embeddings*

Na Figura 6.2 relatamos o boxplot dos tempos levado para os usuários interagirem com os quatro bots, dado em horas. Podemos notar que foi gasto menos tempo nos bots 1, 3 e 4 em comparação ao bot de recomendação aleatória, sendo que o bot 4 possui a menor média de todos. A média está representada pelo símbolo \times no gráfico.

Na Figura 6.3, relatamos a quantidade de mensagens enviadas para cada bot. Essa estatística pode indicar o uso das recomendações em comparação a mensagens escritas. Percebe-se que a recomendação mais utilizada, e que consegue melhor diminuir o número de mensagens escritas, é a do Bot 4, com *embeddings*. O bot aleatório vem em segundo lugar nesse quesito, com o bot de árvore em último.

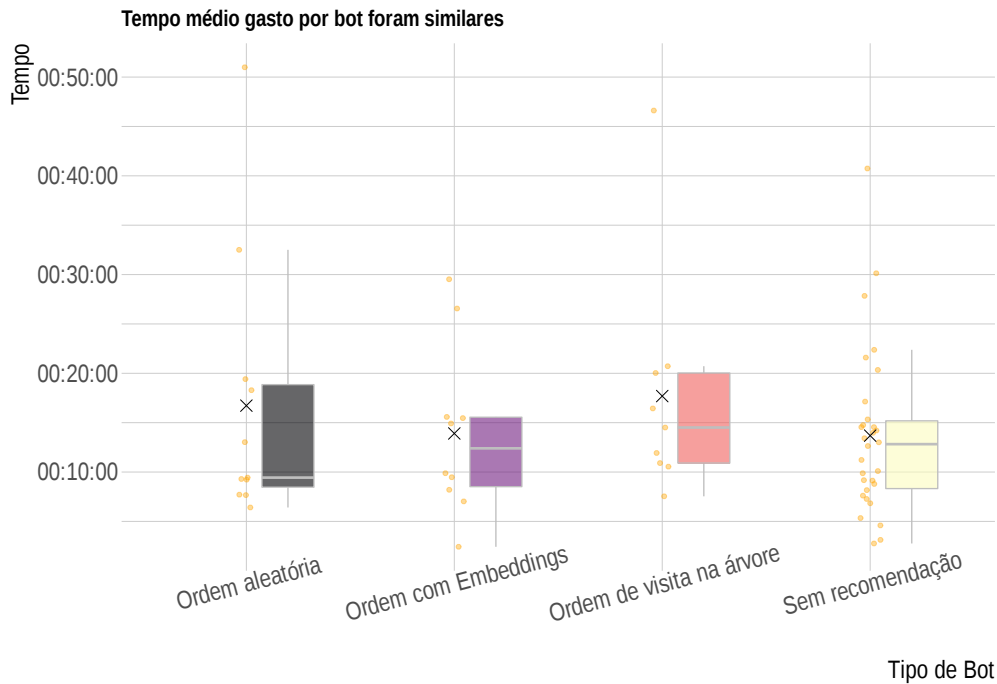


Figura 6.2: Tempo gasto pelos usuários em cada bot

6.3 Avaliação dos Bots (com e sem recomendações)

Os resultados do questionário quanto à avaliação dos bots estão representados nas Figuras 6.4 e 6.5.

Sobre a facilidade em aprender a utilizar o bot, a maioria selecionou concordância com a pergunta, independente de ter ou não recomendação.

Já sobre a facilidade e rapidez no uso, obtivemos uma diferença maior, com ambos tendo mais pontuações de concordância na avaliação do bot com recomendação. No gráfico vemos uma diminuição no grau 5, e aumento nos graus 6 e 7 para o bot com recomendação.

Para a questão “O bot me permitiu fazer boas consultas”, observamos no gráfico uma grande diferença entre os dois tipos de bots. Para o bot com recomendação, uma maioria votou em concordância com a questão, indicando ou um possível benefício das recomendações na realização das consultas, ou que as recomendações são boas consultas por si só. Para o bot sem recomendação no entanto, os participantes responderam em maioria nos graus 4 e 6, indicando uma maior divisão entre a opinião dos participantes.

Para a questão “Achei fácil obter do bot o que queria”, novamente vemos no bot com recomendação uma concentração maior na concordância. O Bot sem recomendação apresenta uma maior dispersão entre os graus 4 e 6. Para as perguntas que avaliam se interagir com bot é trabalhoso, ou até frustrante,

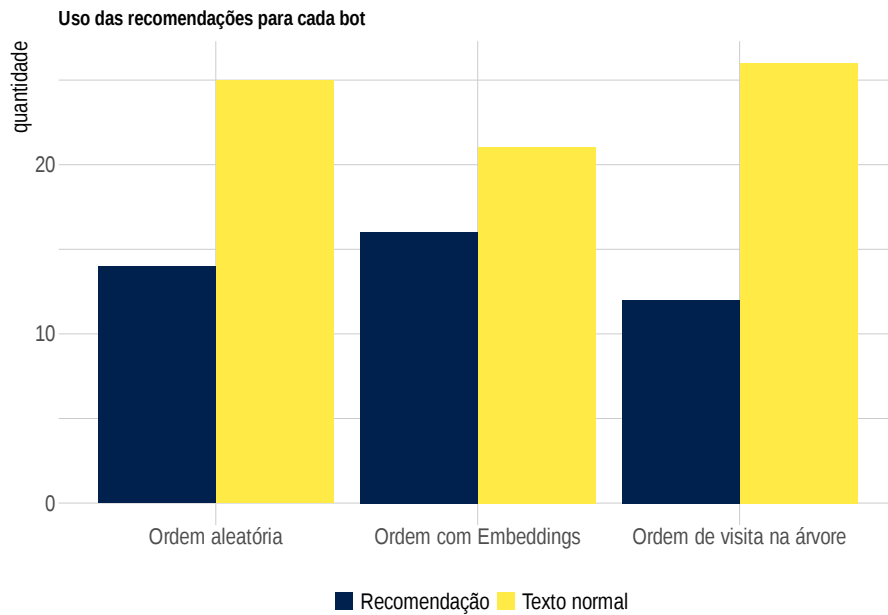


Figura 6.3: Quantidade de mensagem por bot

obtivemos resultados que discordam com a frase em questão em ambos os tipos de bots. No entanto, percebemos no gráfico índices ligeiramente melhores para o bot com recomendação, indicando que possivelmente o uso de recomendação alivia um pouco o esforço de interação com os bots.

Por fim, a sentença “Eu me sinto confiante em buscar informações no bot” obteve resultados bem similares em ambos os tipos de bots, com uma maior dispersão para os bots com recomendação.

6.4 Avaliação das Recomendações

Para avaliar os bots de recomendação, fizemos as seguintes afirmações aos participantes :

- As recomendações do bot foram fáceis de entender.
- As recomendações eram relacionadas à minha pergunta.
- As recomendações foram importantes para o fluxo da conversa.
- As recomendações me ajudaram a pensar em novas perguntas.

A Figura 6.4 abaixo mostra a avaliação geral para o bots com recomendação. Obtivemos resultados muito promissores quanto ao uso de recomendações nos chatbot, com grande parte concordando com as afirmações em questão.

Observamos na Figura 6.6 as respostas aos questionários dos *bots* com recomendação individualmente.

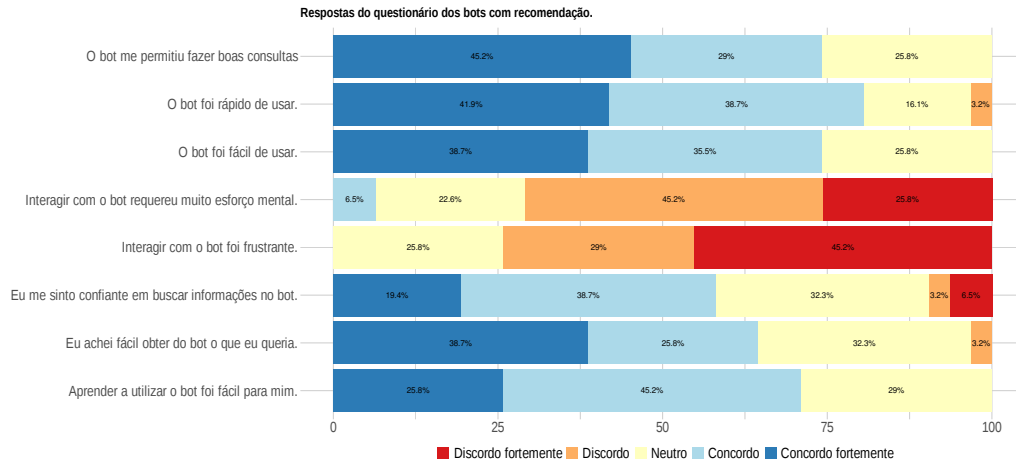


Figura 6.4: Questionário dos bots com recomendação

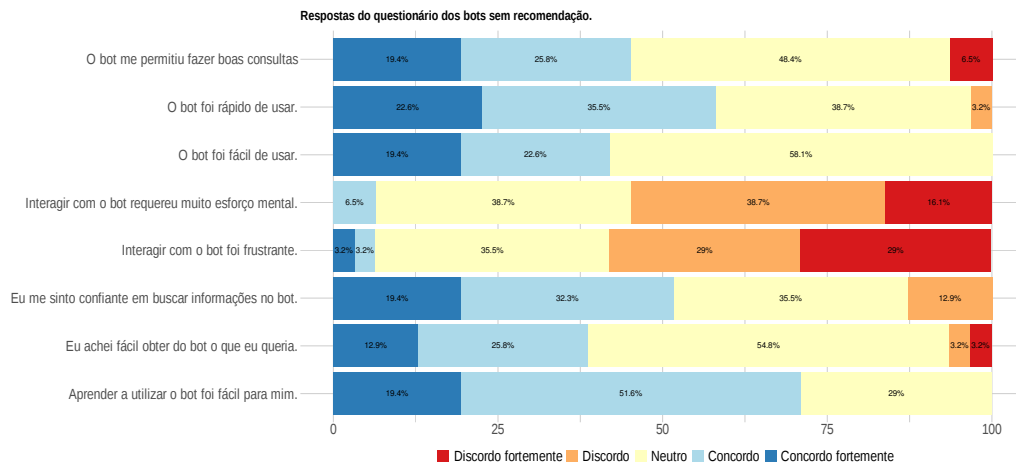


Figura 6.5: Questionário dos bots sem recomendação

Os resultados discutidos neste capítulo nos mostram que as recomendações foram avaliadas positivamente pelos participantes. Entretanto, notamos nas Figuras 6.4 e 6.5 que os usuários se sentiram pouco confiantes na busca por informações *bot*.

Parte dos comentários se referem a limitações do *dataset* utilizado, como falta de informações que poderiam ajudar na compreensão dos resultados. Entretanto, no questionário sobre as recomendações Figura 6.6, percebemos que foram bem aceitas. No próximo capítulo apresentamos as conclusões desse estudo e possíveis trabalhos futuros.

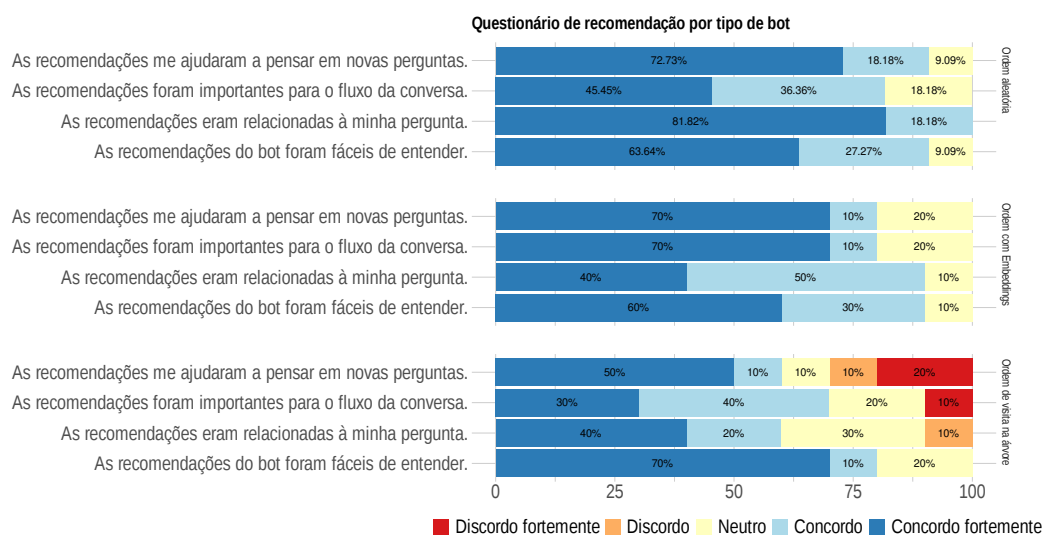


Figura 6.6: Avaliação das recomendações por bot.

7

Conclusões e Trabalhos Futuros

Nesta dissertação propusemos o desenvolvimento de um sistema de recomendação baseado em ontologia, para o qual utilizamos como cenário a ontologia de filmes do IMDb. Discutimos neste capítulo as contribuições e limitações do trabalho realizado, além de possíveis trabalhos futuros.

7.1

Contribuições

Nossa principal contribuição nesta dissertação foi o desenvolvimento de um *bot* que utiliza recomendações com base na hierarquia das propriedades da ontologia. Primeiramente extraímos as relações entre as entidades identificadas na pergunta. Em seguida, fazemos uso delas com intuito de extrair propriedades ou classes relacionadas que, ao serem substituídas, mantenham um sentido semântico.

No questionário de avaliação recebemos várias críticas positivas a respeito das recomendações, conforme descrevemos no capítulo anterior. Alguns usuários relataram que elas facilitaram o uso do sistema por se evitar digitar todo momento a sentença desejada. Uma das reclamações dos usuários foi o fato de o *bot* sugerir algo perguntado anteriormente. Para esta limitação pretendemos inserir um conceito de memória em nosso sistema, no qual perguntas recentes não serão reinseridas como sugestão de pergunta relacionada.

7.2

Trabalhos futuros

No decorrer do desenvolvimento desta pesquisa, identificamos algumas limitações e possíveis melhorias que podem ser feitas.

Primeiramente, apesar de termos conseguido uma média de 10 usuários por *bot* de recomendação, não conseguimos medir com significância estatística qual dessas estratégias teve maior impacto na usabilidade do sistema. Assim, pretendemos conduzir outro estudo, onde os participantes comparam as três estratégias e avaliam qual delas tem melhor performance.

Com o objetivo de diminuir a necessidade de o usuário ter que reescrever informações contidas nos resultados, pretendemos também permitir que o

usuário faça novas perguntas sobre eles, ou seja, perguntas diretamente relacionadas ao conteúdo retornado. Além de possibilitar perguntas sobre um dos elementos contidos na lista de resultados, ao se referenciar diretamente como “quais as atrizes do décimo sétimo filme desse resultado?”

Como mencionamos anteriormente, a nossa recomendação requer que a ontologia possua todas as propriedades contendo *domains* e *ranges*. Por desejarmos experimentar essa abordagem em outras bases de conhecimento, pretendemos explorar estratégias para lidar com essas informações faltantes a fim de superar tal limitação.

Um módulo de *NLG* também pode ser incorporado para maior naturalidade nas respostas do *IQA*, como por exemplo, apresentar informações de forma parcial, permitindo que o usuário realize complementações sobre a resposta. Outra possibilidade é tentar abordagens de recomendações que considerem o grafo de conhecimento, permitindo que o usuário possa ter conhecimento sobre nós relacionados na base.

Por fim, expandir o *bot* para outras plataformas além do *Telegram*, como *Messenger*, ou disponibilizar uma versão *web*, de modo que o usuário não precise se limitar a um *app* de mensagens. Desse modo, poderíamos também testar com um quantidade maior de voluntários.

Referências bibliográficas

- [1] LING, W.; DYER, C.; BLACK, A. W. ; TRANCOSO, I.. **Two/too simple adaptations of word2vec for syntax problems**. In: PROCEEDINGS OF THE 2015 CONFERENCE OF THE NORTH AMERICAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: HUMAN LANGUAGE TECHNOLOGIES, p. 1299–1304, 2015.
- [2] BRIN, S.; PAGE, L.. **The anatomy of a large-scale hypertextual web search engine**. Computer networks and ISDN systems, 30(1-7):107–117, 1998.
- [3] PAGE, L.; BRIN, S.; MOTWANI, R. ; WINOGRAD, T.. **The pagerank citation ranking: Bringing order to the web**. Technical report, Stanford InfoLab, 1999.
- [4] SOUZA, R. R.; ALVARENGA, L.. **A web semântica e suas contribuições para a ciência da informação**. Ciência da Informação, 33(1), 2004.
- [5] BERNERS-LEE, T.; HENDLER, J. ; LASSILA, O.. **The semantic web. a new form of web content that is meaningful to computers will unleash a revolution of new possibilities**. Scientific American, 284(5):3, 2001.
- [6] MORAIS, E. A. M.; AMBRÓSIO, A. P. L.. **Ontologias: conceitos, usos, tipos, metodologias, ferramentas e linguagens**. Relatório Técnico–RT-INF-001/07, dez, 2007.
- [7] GRUBER, T. R.. **Toward principles for the design of ontologies used for knowledge sharing?** International journal of human-computer studies, 43(5-6):907–928, 1995.
- [8] BREITMAN, K. K.. **Web semântica: a internet do futuro**. Grupo Gen-LTC, 2000.
- [9] KIRYAKOV, A.; OGNANOV, D. ; MANOV, D.. **Owlim—a pragmatic semantic repository for owl**. In: INTERNATIONAL CONFERENCE

- ON WEB INFORMATION SYSTEMS ENGINEERING, p. 182–192. Springer, 2005.
- [10] VIEIRA, R.; LOPES, L.. **Processamento de linguagem natural e o tratamento computacional de linguagens científicas**. EM CORPORA, p. 183, 2010.
- [11] DO LAGO PEREIRA, S.. **Processamento de linguagem natural**, 2019. visited 3-Oct-2019.
- [12] AUER, S.; BIZER, C.; KOBILAROV, G.; LEHMANN, J.; CYGANIAK, R. ; IVES, Z.. **Dbpedia: A nucleus for a web of open data**. In: THE SEMANTIC WEB, p. 722–735. Springer, 2007.
- [13] VRANDEČIĆ, D.; KRÖTZSCH, M.. **Wikidata: a free collaborative knowledge base**. 2014.
- [14] DANIEL JURAFSKY, J. H. M.. **Speech and language processing**, Outubro 2019. [Online; postado 02-Outubro-2019].
- [15] GAO, J.; GALLEY, M.; LI, L. ; OTHERS. **Neural approaches to conversational ai**. Foundations and Trends® in Information Retrieval, 13(2-3):127–298, 2019.
- [16] CHEN, H.; LIU, X.; YIN, D. ; TANG, J.. **A survey on dialogue systems: Recent advances and new frontiers**. Acm Sigkdd Explorations Newsletter, 19(2):25–35, 2017.
- [17] CALVANESE, D.; COGREL, B.; KOMLA-EBRI, S.; KONTCHAKOV, R.; LANTI, D.; REZK, M.; RODRIGUEZ-MURO, M. ; XIAO, G.. **Ontop: Answering sparql queries over relational databases**. Semantic Web, 8(3):471–487, 2017.
- [18] LIDDY, E. D.. **Natural language processing**. 2001.
- [19] GRISHMAN, R.; SUNDHEIM, B.. **Message understanding conference-6: A brief history**. In: COLING 1996 VOLUME 1: THE 16TH INTERNATIONAL CONFERENCE ON COMPUTATIONAL LINGUISTICS, 1996.
- [20] BACH, N.; BADASKAR, S.. **A review of relation extraction**. Literature review for Language and Statistics II, 2:1–15, 2007.
- [21] ALLAM, A. M. N.; HAGGAG, M. H.. **The question answering systems: A survey**. International Journal of Research and Reviews in Information Sciences (IJRRIS), 2(3), 2012.

- [22] DIMITRAKIS, E.; SGONTZOS, K. ; TZITZIKAS, Y.. **A survey on question answering systems over linked data and documents.** Journal of Intelligent Information Systems, p. 1–27, 2019.
- [23] SUCHANEK, F. M.; KASNECI, G. ; WEIKUM, G.. **Yago: a core of semantic knowledge.** In: PROCEEDINGS OF THE 16TH INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, p. 697–706, 2007.
- [24] KONSTANTINOVA, N.; ORASAN, C.. **Interactive question answering.** In: EMERGING APPLICATIONS OF NATURAL LANGUAGE PROCESSING: CONCEPTS AND NEW RESEARCH, p. 149–169. IGI Global, 2013.
- [25] QIN, L.; HAO, Z. ; YANG, L.. **Question answering system based on food spot-check knowledge graph.** In: PROCEEDINGS OF 2020 THE 6TH INTERNATIONAL CONFERENCE ON COMPUTING AND DATA ENGINEERING, p. 168–172, 2020.
- [26] LI, Y.; CAO, J. ; WANG, Y.. **Implementation of intelligent question answering system based on basketball knowledge graph.** In: 2019 IEEE 4TH ADVANCED INFORMATION TECHNOLOGY, ELECTRONIC AND AUTOMATION CONTROL CONFERENCE (IAEAC), volumen 1, p. 2601–2604. IEEE, 2019.
- [27] KULKARNI, A.; MEHTA, K.; GARG, S.; BANSAL, V.; RASIWASIA, N. ; SENGAMEDU, S.. **Productqna: Answering user questions on e-commerce product pages.** In: COMPANION PROCEEDINGS OF THE 2019 WORLD WIDE WEB CONFERENCE, p. 354–360, 2019.
- [28] KADARIYA, D.; VENKATARAMANAN, R.; YIP, H. Y.; KALRA, M.; THIRUNARAYANAN, K. ; SHETH, A.. **kbot: Knowledge-enabled personalized chatbot for asthma self-management.** In: 2019 IEEE INTERNATIONAL CONFERENCE ON SMART COMPUTING (SMARTCOMP), p. 138–143. IEEE, 2019.
- [29] AVILA, C.; CALIXTO, A.; ROLIM, T.; FRANCO, W.; VENCESLAU, A.; VIDAL, V.; PEQUENO, V. ; MOURA, F.. **Medibot: An ontology based chatbot for portuguese speakers drug’s users.** In: INTERNATIONAL CONFERENCE ON ENTERPRISE INFORMATION SYSTEMS, volumen 1, p. 25–36. SciTePress, 2019.
- [30] MITTAL, A.; SEN, J.; SAHA, D. ; SANKARANARAYANAN, K.. **An ontology based dialog interface to database.** In: PROCEEDINGS

- OF THE 2018 INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, p. 1749–1752, 2018.
- [31] LE-HONG, P.; BUI, D.-T.. **A factoid question answering system for vietnamese**. In: COMPANION PROCEEDINGS OF THE THE WEB CONFERENCE 2018, p. 1049–1055, 2018.
- [32] ESHGHISHARGH, A.; GRAY, K.; MILTON, S. K. ; KOLBE, S. C.. **A semantic system for answering questions in neuroinformatics**. In: PROCEEDINGS OF THE AUSTRALASIAN COMPUTER SCIENCE WEEK MULTICONFERENCE, p. 1–5, 2018.
- [33] SANTOSO, H. A.; WINARSIH, N. A. S.; MULYANTO, E.; SUKMANA, S. E.; RUSTAD, S.; ROHMAN, M. S.; NUGRAHA, A.; FIRDAUSILLAH, F. ; OTHERS. **Dinus intelligent assistance (dina) chatbot for university admission services**. In: 2018 INTERNATIONAL SEMINAR ON APPLICATION FOR TECHNOLOGY OF INFORMATION AND COMMUNICATION, p. 417–423. IEEE, 2018.
- [34] ABDI, A.; IDRIS, N. ; AHMAD, Z.. **Qapd: an ontology-based question answering system in the physics domain**. *Soft Computing*, 22(1):213–230, 2018.
- [35] WANNER, L.; ANDRÉ, E.; BLAT, J.; DASIOPOULOU, S.; FARRÙS, M.; FRAGA, T.; KAMATERI, E.; LINGENFELSER, F.; LLORACH, G.; MARTÍNEZ, O. ; OTHERS. **Kristina: A knowledge-based virtual conversation agent**. In: INTERNATIONAL CONFERENCE ON PRACTICAL APPLICATIONS OF AGENTS AND MULTI-AGENT SYSTEMS, p. 284–295. Springer, 2017.
- [36] TONIUC, D.; GROZA, A.. **Climobot: An argumentative agent for climate change**. In: 2017 13TH IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT COMPUTER COMMUNICATION AND PROCESSING (ICCP), p. 63–70. IEEE, 2017.
- [37] KINCAID, R.; POLLOCK, G.. **Nicky: Toward a virtual assistant for test and measurement instrument recommendations**. In: 2017 IEEE 11TH INTERNATIONAL CONFERENCE ON SEMANTIC COMPUTING (ICSC), p. 196–203. IEEE, 2017.
- [38] GARRIDO, P.; BARRACHINA, J.; MARTINEZ, F. ; SERON, F.. **Smart tourist information points by combining agents, semantics and**

- ai techniques. *Computer Science and Information Systems*, 14(1):1–23, 2017.
- [39] DEVI, M.; DUA, M.. **Adans: An agriculture domain question answering system using ontologies**. In: 2017 INTERNATIONAL CONFERENCE ON COMPUTING, COMMUNICATION AND AUTOMATION (ICCCA), p. 122–127. IEEE, 2017.
- [40] ALBARGHOTHI, A.; KHATER, F. ; SHAALAN, K.. **Arabic question answering using ontology**. *Procedia Computer Science*, 117:183–191, 2017.
- [41] BOCKLISCH, T.; FAULKNER, J.; PAWLOWSKI, N. ; NICHOL, A.. **Rasa: Open source language understanding and dialogue management**. arXiv preprint arXiv:1712.05181, 2017.
- [42] OTTER, D. W.; MEDINA, J. R. ; KALITA, J. K.. **A survey of the usages of deep learning for natural language processing**. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [43] SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I. ; SALAKHUTDINOV, R.. **Dropout: a simple way to prevent neural networks from overfitting**. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [44] DEVLIN, J.; CHANG, M.-W.; LEE, K. ; TOUTANOVA, K.. **Bert: Pre-training of deep bidirectional transformers for language understanding**. arXiv preprint arXiv:1810.04805, 2018.

A

Questionário de avaliação

Apresentamos o questionário de avaliação utilizado para avaliar os bots com e sem recomendação. A diferença entre os questionários é a presença da seção **Recomendações** para avaliação dos bots que implementam essa estratégia.

Questionário de Avaliação - Ontology IQA Bot (com recomendação)

Este questionário tem como objetivo permitir coletar informações sobre a experiência do usuário ao utilizar o nosso bot de filmes.

Este estudo faz parte da pesquisa de Mestrado de Jessica Paloma Sousa Cardoso, sob orientação da Profa. Simone D.J. Barbosa, do Departamento de Informática da PUC-Rio. Os dados coletados serão utilizados apenas no desenvolvimento desta pesquisa e não serão utilizados como forma de identificação dos seus participantes.

Ao prosseguir, você estará consentindo em compartilhar os dados coletados para essa pesquisa. Ao prosseguir, você estará consentindo em compartilhar os dados coletados para essa pesquisa.

***Obrigatório**

1. Informe seu ID do Telegram *

2. Você faz parte de qual grupo do experimento? *

Marcar apenas uma oval.

- Grupo 1
- Grupo 2
- Grupo 3
- Grupo 4
- Grupo 5
- Grupo 6

3. Qual a sua faixa etária? *

Marcar apenas uma oval.

- De 17 a 24 anos
- De 25 a 34 anos
- De 35 a 44 anos
- De 45 a 54 anos
- Mais de 54 anos

4. Qual é o seu grau de escolaridade? *

Marcar apenas uma oval.

- Fundamental incompleto
- Fundamental completo
- Médio incompleto
- Médio completo
- Superior incompleto
- Superior completo
- Cursando pós-graduação
- Pós-graduado

5. Campo de atuação

6. Você já utilizou algum serviço de chatbot anteriormente? *

Marcar apenas uma oval.

Sim

Não

7. Qual seu nível de familiaridade com chatbots em geral? *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Nenhum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Especialista

8. Você costuma usar o Telegram com que frequência? *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Frequentemente

9. Qual plataforma você utilizou para acessar o chatbot? *

Marcar apenas uma oval.

Desktop

Web

Mobile

Questionário sobre o Bot

10. Aprender a utilizar o bot foi fácil para mim. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo

11. O bot foi fácil de usar. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo

12. O bot foi rápido de usar. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo

13. O bot entendeu o que eu escrevi. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo

14. O bot me permitiu fazer boas consultas. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo

15. O bot me permitiu ter controle sobre o que consulto. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo

16. As respostas do bot foram fáceis de entender. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo

17. As respostas do bot foram condizentes com o que perguntei. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo

18. Eu achei fácil obter do bot o que eu queria. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo

19. Interagir com o bot requereu muito esforço mental. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo

20. Interagir com o bot foi frustrante. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo

21. Eu me sinto confiante em buscar informações no bot. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo

Recomendações

22. As recomendações do bot foram fáceis de entender. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo

23. As recomendações eram relacionadas à minha pergunta. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo

24. As recomendações foram importantes para o fluxo da conversa. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo

25. As recomendações me ajudaram a pensar em novas perguntas. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo

Comentários finais

26. Você gostaria de fazer mais algum comentário, crítica ou sugestão sobre o bot?

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

INSTRUÇÕES

O OntoQA-Bot faz parte da pesquisa de mestrado de Jessica Cardoso, aluna do Departamento de Informática da PUC-Rio, sob orientação da Profa. Simone Diniz Junqueira Barbosa. Você está sendo convidado para participar de um estudo que consiste em interagir com um chatbot e fornecer sua opinião sobre essa interação. Sua participação é totalmente voluntária, e os dados serão coletados de forma a assegurar sua privacidade. Nenhum dado será publicado de forma que você ou qualquer outro participante possa ser identificado.

Nessa pesquisa, além das respostas aos questionários, vamos coletar as conversas que você travar com o nosso chatbot, além do seu id e nome de usuário.

Domínio da conversa

Este bot responde estritamente perguntas relacionadas ao domínio de filmes e séries dos diretores Scorsese e Tarantino cadastrados no IMDb, permitindo a consulta aos seguintes atributos:

- **Staff de filmes e séries:** Ator, Atriz, Diretor, Produtor, Escritor/Roteirista, Figurinista
- **Propriedades de filmes/séries:** prêmio, indicação, data de indicação, local de filmagem, gênero, data de estreia, duração, orçamento, receita bruta, ano de produção, companhia de produção.

O gênero pode ser decomposto nas seguintes categorias:

- **Entretenimento:** Diversão, Amor, Muito Sensível, Emocionante Sensível, Emocionante Lógico, Ficção Científica e Fantasia, Musical (Entretenimento Musical), Informação de Entretenimento e Ação Bruta
- **Informação:** Informação Televisiva e Informativo Histórico

Propriedades disponíveis sobre as pessoas desse domínio::

- data de nascimento, data de indicação, indicações, premiações.

Prêmios cadastrados:

- Oscar
- BAFTA
- Emmy
- Diverso
- Globo de ouro

Observações:

- Pode ser possível trocar os termos por palavras equivalentes, por exemplo, a troca do termo ação bruta por violência.
- Os filmes estão com seus títulos originais, muitos em inglês.
- Os nomes próprios devem estar com as suas primeiras letras maiúsculas (nome de pessoa, filme ou série)
 - Four Rooms
 - Robin Johnson
 - Jimmy Kimmel Live!
- Operações de quantidade, comparações ou intervalo entre valores não são cobertas pelo bot.
- O bot responde apenas a data inteira, sem filtragem por dia, mês, ano ou idade.
- Se você não souber nomes de filmes na sua forma original (já que muitos estão em inglês), é recomendável começar perguntando por filmes de um dado gênero, ou que ganharam um prêmio específico.
 - “Me liste filmes do gênero diversão”
 - “quais os filmes que ganharam um oscar”

Regras de uso

- O chatbot responde somente perguntas simples e factuais, sobre o domínio descrito anteriormente.
- Os dois tipos de perguntas cobertas são:
 - **ask:** perguntas com resposta 'sim' ou 'não'. Ex: “*O filme After Hours obteve algum prêmio?*”
 - **select:** perguntas que retornam uma lista de resultados. Ex: “*Fale-me os prêmios do filme After Hours*”
 - Ele **não** responde perguntas complexas, como por exemplo:
 - perguntas compostas: atores e atrizes do filme After Hours
 - perguntas específicas: filmes de diversão com orçamento maior que x, atores de filmes do gênero diversão,
- **Sempre que possível definir o tipo do que se pergunta.**
 - **Ator, Atriz, Escritor, etc, antes do nome da pessoa**
 - **Filme, Série, antes do título**
 - **Gênero antes do subtipo de gênero**
 - **Empresa antes do nome da produtora**
 - Por exemplo:
 - filmes com a atriz Robin Johnson
 - filmes do gênero diversão
 - orçamento do filme After Hours
 - Não é possível perguntar sobre os resultados retornados, apenas sobre as perguntas anteriores.