UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
DEPARTAMENTO DE INFORMÁTICA
MESTRADO EM INFORMÁTICA

MARTA TALITHA CARVALHO FREIRE DE AMORIM

UM SISTEMA INTELIGENTE BASEADO EM ONTOLOGIA PARA APOIO AO ESCLARECIMENTO DE DÚVIDA

VITÓRIA, AGOSTO 2012

MARTA TALITHA CARVALHO FREIRE DE AMORIM

UM SISTEMA INTELIGENTE BASEADO EM ONTOLOGIA PARA APOIO AO ESCLARECIMENTO DE DÚVIDA

Dissertação submetida Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Mestre em Informática.

VITÓRIA, AGOSTO 2012

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

Amorim, Marta Talitha Carvalho Freire de, 1982-
A524s Um sistema inteligente baseado em ontologia para apoio ao esclarecimento de dúvida / Marta Talitha Carvalho Freire de Amorim. – 2012.
89 f. : il.

Orientador: Davidson Cury.
Co-Orientador: Crediné Silva de Menezes.
Dissertação (Mestrado em Informática) – Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Sistemas de consultas e respostas. 2. Ontologia. 3. Recuperação da informação. I. Cury, Davidson. II. Menezes, Crediné Silva de, 1952-. III. Universidade Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 004

MARTA TALITHA CARVALHO FREIRE DE AMORIM

UM SISTEMA INTELIGENTE BASEADO EM ONTOLOGIA PARA APOIO AO ESCLARECIMENTO DE DÚVIDA

Dissertação submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Mestre em Informática.

Aprovada em 31 de Agosto de 2012.

COMISSÃO EXAMINADORA

_____

Prof. Dr. Davidson Cury
Universidade Federal do Espírito Santo (UFES)
(Orientador)

_____

Prof. Dr. Crediné Silva Menezes
Universidade Federal do Espírito Santo (UFES)
(Co-orientador)

_____

Prof. Dr. Orivaldo de Lira Tavares
Universidade Federal do Espírito Santo (UFES)

_____

Prof. Dr. Alexandre Ibrahim Direne
Universidade Federal do Paraná (UFPR)

VITÓRIA, AGOSTO 2012

DEDICATÓRIA

Dedico este trabalho à

A El Shaddai, meu criador, sustentador e minha força.

AGRADECIMENTOS

Agradeço a Deus, por estar comigo durante essa caminhada, me iluminando e concedendo graça para concluir este trabalho.

Agradeço ao meu orientador professor Dr. Davidson Cury pelo incentivo, pela disponibilidade e pela amizade. Obrigada por me motivar e ter confiado no meu trabalho.

Outro agradecimento especial eu dedico ao professor Dr. Crediné, meu co-orientador pelo seu apoio e sugestões.

Aos professores membros da Banca Examinadora, por terem atendido ao convite para desempenhar este papel, dispondo de seu tempo e conhecimento para analisar este trabalho.

Ao meu querido esposo Warley Rocha Mendes pelo amor, compreensão e por toda ajuda ao longo deste trabalho.

Aos meus pais Gerson Freire de Amorim Filho e Rosangela Carvalho pelo incentivo, pelas orações, carinho e amor. A minha irmã Isabela Carvalho Freire de Amorim pelo incentivo. A minha irmã Evelin Carvalho Freire de Amorim que muito me apoiou com seus conselhos e trocas de ideias.

Aos queridos professores Heráclito Amâncio Pereira Junior e Elizabeth Maria Klippel por me apoiar.

Aos meus amigos do LIED, Ernani Leite, Pedro David, Ewerton Bada, Maiksson Baldan, Erick Sperandio, Wilson Guasti e Carlos Alexandre por compartilharem dificuldades e conquistas na realização deste curso.

Ao amigo Sergio Teixeira que disponibilizou seu tempo e conhecimento.

Ao professor Dr. José Manuel Gómez Soriano que embora estando tão longe (na Espanha), colaborou com esta pesquisa de forma a esclarecer muitas dúvidas. Agradeço a atenção oferecida.

As pessoas incríveis do Nemo com quem pude trocar muitas ideias, Roberto Carrareto, Veruska Zamborlini, Pedro Paulo Favato, Victor Viola e Paulo César Fernandes.

Aos colegas de trabalho, especialmente àqueles que sempre se disponibilizaram a acertos necessários de horário para que eu pudesse cumprir com todas as minhas obrigações profissionais e acadêmicas.

A todos os meus amigos e amigas que, direta ou indiretamente, sempre estiveram presentes me aconselhando e incentivando.

" O homem pode tanto quanto sabe "

Francis Bacon

RESUMO

Quando as pessoas querem aprender algum conceito, a forma mais comum é usar uma ferramenta de pesquisa, como: Google, Yahoo, Bing, dentre outros. Uma consulta em linguagem natural é submetida para uma ferramenta e a pesquisa retorna uma grande quantidade de páginas relacionadas ao conceito pesquisado. Geralmente as páginas retornadas são listadas e organizadas principalmente baseando-se na combinação de palavras chaves ao invés de utilizar a interpretação e a relevância dos termos consultados. O usuário terá que ler uma grande quantidade de páginas e selecionar a mais apropriada a sua necessidade. Esse tipo de comportamento consome tempo e o foco do usuário-aprendiz é disperso do seu objetivo.

A utilização de um sistema inteligente que apoie o esclarecimento de dúvidas pretende resolver esse problema, apresentando as respostas mais precisas ou frases para as perguntas em linguagem natural. Exemplos de sistemas de esclarecimento de dúvidas são: sistema de pergunta-resposta, *help-desk* inteligentes, entre outros.

Este trabalho utiliza uma abordagem arquitetônica para um sistema de pergunta-resposta baseado em três passos: análise da pergunta, seleção e extração da resposta e geração da resposta. Um dos méritos dessa arquitetura é utilizar técnicas que se complementam, tais como: ontologias, técnicas de recuperação de informação e uma base de conhecimento escrita em linguagem AIML para extrair a resposta de forma rápida. O foco deste trabalho é responder perguntas *WH-question* (O que, Quem, Quando, Onde, Quais, Quem) da língua inglesa.

Palavras-chave: Sistema de Pergunta-Resposta, Ontologia, e Recuperação da Informação.

ABSTRACT

When people want to learn a concept, the most common way is to use a search engine like: Google, Yahoo, Bing, among others. A natural language query is submitted to a search tool and which returns a lot of pages related to the concept studied. Usually the returned pages are listed and organized mainly based on the combination of keywords instead of using the interpretation and relevance of the terms found. The user must have read a lot of pages and selects the most appropriate to his needs. This kind of behavior takes time and focus on user-learner is dispersed to his goal.

The use of intelligent systems that support the clarification of doubt has intent to solve this problem, presenting the most accurate answers to questions or sentences in natural language. Examples clarification of doubt systems are: question-answer system, help-desk intelligent among others.

This work uses an architectural approach to a question answering system based on three steps: question analysis, selection and extraction of the answer and answer generation. One of the merits of this architecture is to use techniques that complement each other, such as ontologies, information retrieval techniques and a knowledge base written in AIML language to extract the answer quickly. The focus of this work is to answer questions WH-question (What, Who, When, Where, What, Who) of the English language.

Keywords: Question Answering System, Ontology and Information Retrieval.

SUMÁRIO DE FIGURAS

SUMÁRIO DE TABELAS

SUMÁRIO

CAPÍTULO 1 INTRODUÇÃO

Um sistema de pergunta-resposta é definido como uma tarefa por intermédio da qual uma máquina automatizada (tal como um computador) responde às perguntas arbitrárias formuladas em linguagem natural. Sistemas de pergunta-resposta[1] são especialmente úteis em situações na qual o usuário precisa saber uma parte muito específica de informação e não tem tempo – ou apenas não quer – ler toda documentação disponível relacionada ao tópico pesquisado para resolver o problema (VICEDO *et al.*, 2007).

A partir do teste de Turing[2], sistema de pergunta-resposta tem sido frequentemente usado como forma direta de observar e medir o comportamento inteligente em máquinas. Apesar da inteligência artificial (IA) ter se diversificado muito além da noção de inteligência proposta pelo teste de Turing, sistemas de pergunta-resposta permanecem com as principais competências necessárias para uma grande gama de classes de sistemas. Os problemas resolvidos pelos sistemas de pergunta-resposta se estendem além da inteligência artificial para muitas tarefas analíticas que envolvem recuperação, correlação e análise de informação de maneira que podem naturalmente ser formuladas como perguntas (GUNNING *et al.*, 2010).

Dentro da competência de inteligência artificial sistemas de pergunta-resposta têm sido abordados a partir de diferentes perspectivas. Abordagens baseadas nas ciências cognitivas estão preocupadas com a simulação humana em responder e perguntar. Problemas de compreensão e geração da linguagem natural vêm à tona em sistemas de pergunta-resposta, pois grandes bases de dados de documento requer uma análise linguística sofisticada, incluindo entendimento do discurso e sumarização de texto. Os mecanismos de raciocínio para sistemas de pergunta-resposta são preocupações dos pesquisadores na representação do conhecimento (BURHANS, 2002).

[1] Também comumente chamado na língua inglesa de sistemas *question answering* - QA

[2]Alan Turing em 1950 propôs um teste (conhecido como o teste de Turing) para avaliar o comportamento inteligente em uma máquina. A proposta de Turing era permitir que um ser humano, a que chamamos de interrogador, se comunique com um objeto de teste, por meio de uma máquina de escrever, sem saber se o objeto é uma pessoa ou uma máquina. Nesse ambiente, declarar-se-ia que a máquina tem comportamento inteligente naquela situação se o interrogador não conseguisse distingui-la de um ser humano (BROOKSHEAR, 2011).

Um sistema de pergunta-resposta fornece resposta *exata* para perguntas em linguagem natural para certa variedade de assunto. A noção de *exata* nesse contexto é uma medida subjetiva que pretende indicar que um sistema de pergunta-resposta distingue-se por tentar fornecer a resposta que contém apenas a informação necessária para responder a pergunta. A resposta *exata* pode trazer informações adicionais ou complementares, incluindo uma justificação ou diálogo, explicando o porquê a resposta está correta (FERRUCCI, 2009).

Sistemas de pergunta-reposta buscam oferecer a mesma facilidade que ocorrem nos diálogos entre as pessoas, onde as dúvidas são respondidas prontamente. Eles vão além da busca mais familiar baseada em palavras chaves (como no Google, Yahoo, e entre outros motores de busca), na tentativa de reconhecer o que a pergunta expressa e apresentar uma resposta correta. Isso simplifica para os usuários de duas formas. Primeiro, perguntas não se traduzem em uma simples lista de palavras chaves. E segundo que sistemas pergunta-resposta assumem a responsabilidade de fornecer a resposta ao invés de uma lista de documentos potencialmente relevantes (CLARK *et al.*, 2010).

Existe uma diferença entre busca baseada em palavras-chave, em que são usadas para responder as perguntas e aquelas usadas nos motores de busca. A intenção da primeira é encontrar a resposta à pergunta que é interpretada por meio de um conjunto de palavras-chave, para a última o objetivo é encontrar documentos relevantes que conterão as palavras chaves (DAMIJANOVIC *et al.*, 2010).

Observe a seguinte situação hipotética: Realizar uma consulta no Google com a frase "*Quais vegetais previnem a osteoporose*?" Não encontraremos a resposta (ou de preferência, nenhum documento que contém a resposta). A resposta pode ser encontrada nos documentos disponíveis na Web. Entretanto, motores de busca (ex: Google) podem não localizá-la por não implementarem raciocínio ou inferências lógicas. Isto é, *brócolis* é um vegetal que previne a osteoporose – mas nenhum documento da Web menciona isso. No entanto, existem documentos que mencionam o seguinte: *brócolis é um vegetal* (1), *brócolis contém cálcio* (2), *Cálcio previne osteoporose* (3). Sistemas de recuperação de informação que usam ontologias são construídos para responder estes e tipos similares de perguntas (DAMIJANOVIC *et al.*, 2010).

Tomemos outro exemplo: (1) *Qual país foi visitado pelo Papa em 1960?* , as palavras chaves são: "*país*", "P*apa*", "*visitado*", "1960". Nenhuma dessas palavras denota um país particular (tal como "*Reino Unido*", ou "*Estados Unidos*"), ou "*Papa*" (chefe da igreja católica, por exemplo), ou a data dentro do intervalo de 10 anos entre 1960 e 1970. Um conjunto muito mais complexo de palavras chaves é necessário a fim de se aproximar do resultado pretendido. Experiências mostram que pessoas não aprenderão como formular e usar esse conjunto (CLARK *et al.*, 2010).

A natureza de um domínio específico (restrito) afeta os tipos de perguntas e respostas que podem ser esperadas. Consequentemente, diferentes domínios restritos se beneficiam de diferentes técnicas de resolução da resposta. (VICEDO *et al.*, 2007; MINOCK *et al.*, 2005) enumeram três técnicas desejadas para um domínio restrito dentro do contexto da web. De acordo com Minock, um domínio restrito deve possuir as seguintes características: (1) Deve ser circunscrito, ou seja, o limite do domínio é determinado (2) Deve ser complexo (3) Deve ser prático (VICEDO *et al.*, 2007).

Outro fator importante é identificar os tipos de perguntas que serão tratadas pelo sistema, pois há infinitas maneiras de fazer perguntas. Podemos categorizar perguntas em factoides (representam fatos) ou procedimentais; A resposta para pergunta do tipo factoide é uma frase curta ou uma única palavra. A resposta para perguntas procedimentais pode consistir de sentenças ou parágrafos (KONCHADY, 2008). A seguir, a Figura 1.1 mostra um esquema dos tipos de pergunta:

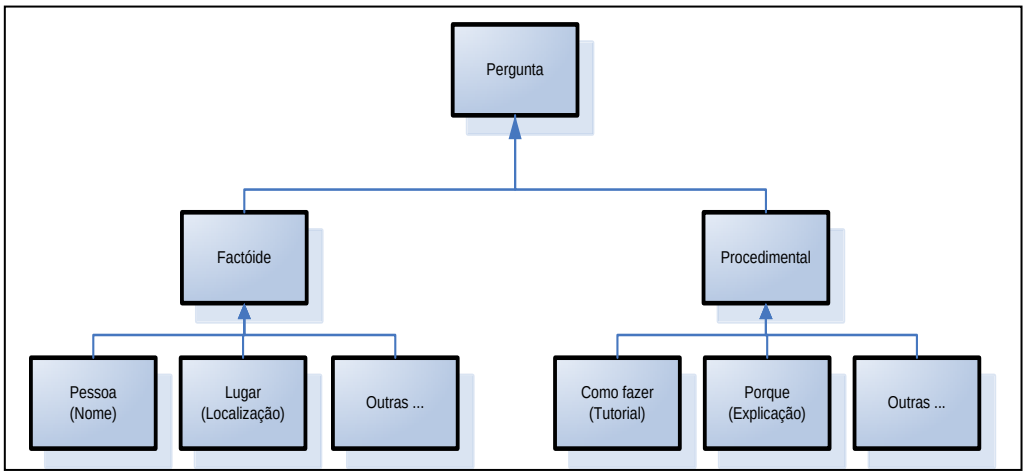

Figura 1.1 – Os tipos de pergunta (Fonte: KONCHADY, 2008)

O desafio de um sistema de pergunta-resposta é retornar a resposta que mais se aproxima do resultado para as perguntas feitas em linguagem natural. O processo completo é bastante complicado, pois requer um número de diferentes técnicas trabalhando em conjunto a fim de atingir o objetivo, incluindo a reescrita e formulação da consulta, classificação da pergunta, recuperação da informação, recuperação de passagens textuais, extração da resposta, ordenação da resposta e justificação (AKERKAR *et al.*, 2009).

A adequação e a simplicidade de uma interface de pergunta-resposta é tão obvia que, desde os primórdios da ficção científica, quase toda história em que computadores estão presentes, eles assumem ser máquinas falantes que, sem esforço, entendem e respondem qualquer pergunta concebível. No entanto, o melhor que experimentamos hoje é o uso inteligente de busca por palavras-chaves para achar um ou mais documentos relacionados. Esperamos máquinas inteligentes capazes de responder perguntas – fornecer respostas específicas pela compreensão, síntese, raciocínio sobre dados e documentos de texto. E que essas máquinas também sejam capazes de fornecer para os usuários explanações ou justificações relacionadas ao contexto da pergunta (GUNNING *et al.*, 2010).

Adicionam-se a essas máquinas, por demanda deste trabalho, a capacidade de apresentar respostas (tenta se aproximar ao máximo da resposta desejada.) ou frases para perguntas do domínio específico de sistemas operacionais. Essa máquina atuará como uma ferramenta facilitadora para a aprendizagem no apoio ao esclarecimento de dúvidas.

1.1 MOTIVAÇÃO

Muitas pesquisas feitas na área dos sistemas de pergunta-resposta têm sido desenvolvidas utilizando-se técnicas diversas (CAO *et al.*, 2011; GLÖCKNER *et al.*, 2009) e abordagens para aprimorar os resultados de *recall*[3] e *precision*[4].

[3]*Recall* ou abrangência é uma medida comumente utilizada por sistemas de recuperação de informação. Recall pode ser definido como o número de documentos relevantes recuperados divididos pelo número de documentos relevantes para uma consulta (BAEZA-YATES *et al.*, 2000).

Iniciando em 1999, a conferência TREC (*Text Retrieval Conference*) foi uma das primeiras tentativas sistemáticas de avaliar os sistemas de pergunta-resposta sobre uma coleção de documentos comum. Inúmeras soluções da área de pergunta-resposta apresentaram resultados relevantes na recuperação da resposta, mas poucas exploraram ontologias de domínio e bases de conhecimento AIML.

A motivação é construir uma nova arquitetura de um sistema de pergunta-resposta que realize uma transformação das perguntas em consultas enriquecidas por uma ontologia de domínio e pela *Wordnet*[5]. Essa transformação é um grande desafio conhecido em processamento de linguagem natural devido às características inatas mais relevantes neste contexto sejam a ambiguidade e a complexidade (CHURCH *et al.*, 1982).

A partir do trabalho de mestrado desenvolvido por Teixeira (TEIXEIRA, 2005), no qual objetivou a construção automática de bases de conhecimento para *chatterbots*, percebemos a oportunidade de extensão para desenvolver um sistema de pergunta-resposta. Dessa forma aproveitamos a base de conhecimento de perguntas e respostas com o propósito de aperfeiçoar a recuperação da pergunta.

1.2 OBJETIVO

O objetivo geral desta dissertação é propor uma nova arquitetura para um sistema de pergunta-resposta, apoiado por várias técnicas distintas e que se complementam, tais como: recuperação de informação, ontologias, processamento de linguagem natural, entre outras, com o fim de aprimorar a busca de respostas.

Para alcançar esse objetivo geral, as seguintes etapas específicas devem ser realizadas:

- Estruturar e caracterizar a arquitetura no nível conceitual, definindo as etapas e regras do sistema;
- Identificar os tipos de perguntas tratadas pelo sistema;

---

[4]*Precision* ou precisão é o número de documentos relevantes recuperados divididos pelo número de documentos recuperados pelo sistema para uma consulta (BAEZA-YATES *et al.*, 2000).

[5]Wordnet – é uma base de conhecimentos linguísticos para a língua inglesa - http://wordnet.princeton.edu/

- Determinar as bases de conhecimento e o domínio abrangido nos experimentos;
- Reconhecer o conhecimento disponível em textos e gerar uma representação formal desse conhecimento na ontologia (extração de Informação e representação de conhecimento);
- Mapear os componentes da arquitetura do nível conceitual para as soluções tecnológicas;

1.3 QUESTÕES NORTEADORAS

De acordo com os objetivos gerais, as indagações que a pesquisa se propõe a responder são:

i. Quais as implicações no uso de ontologias de domínio para enriquecimento da extração da resposta?

ii. Que sentido assume a técnica de implicação textual (RTE) em um sistema de pergunta resposta?

iii. A base de conhecimento AIML pode aperfeiçoar uma arquitetura de um sistema de pergunta-resposta?

1.4 METODOLOGIA

Não há, evidentemente, regras fixas acerca da elaboração de um projeto. Sua estrutura é determinada pelo tipo de problema a ser pesquisado e também pelo estilo de seus autores. É necessário que o projeto esclareça como se processará a pesquisa, quais as etapas que serão desenvolvidas e quais os recursos que devem ser alocados para atingir seus objetivos (GIL, 2002). Em sentido amplo de pesquisa, a investigação científica tratada neste trabalho tem por metodologia as seguintes características:

Tabela 1.1 – Características da metodologia científica aplicada ao trabalho.

Quanto à natureza	Pesquisa aplicada
Quanto aos objetivos	Exploratório e descritiva
Quanto às abordagens	Qualitativa
Quanto aos procedimentos	Experimental

Baseado nestas características o trabalho foi desenvolvido conforme as etapas apresentadas na Figura 1.2. Essas etapas englobam fases, processos, ações principais e produção de artefatos.

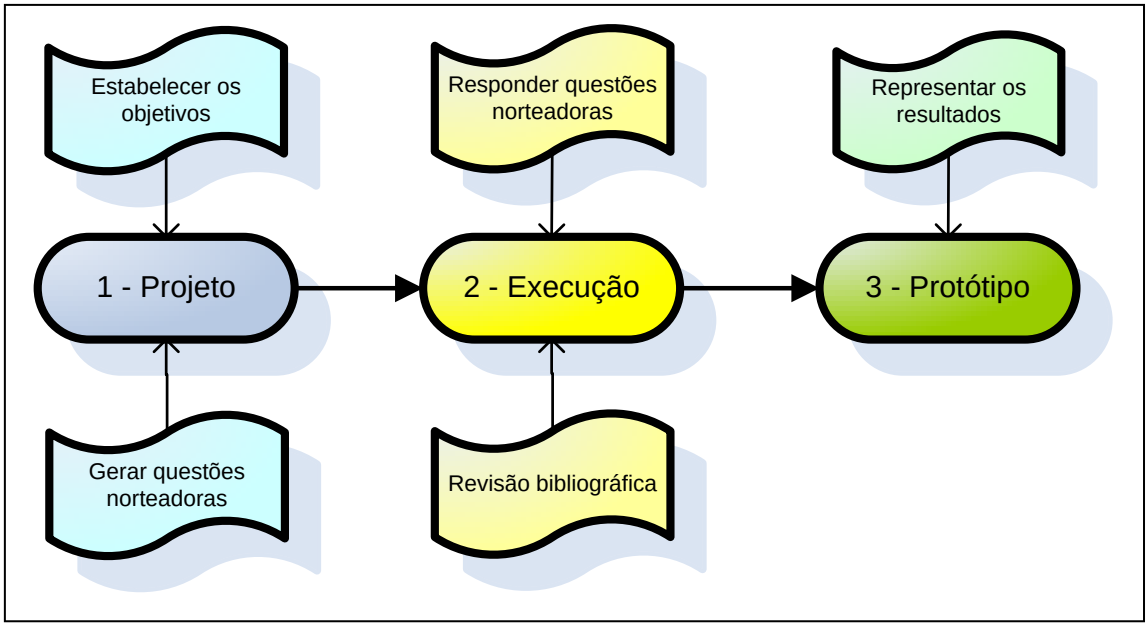

Figura 1.2 – Etapas do planejamento do desenvolvimento da pesquisa.

A primeira fase, "Projeto", preocupa-se em formar os limites da pesquisa e definir os objetivos. A fase seguinte, "Execução", são realizadas a pesquisa bibliográfica, análises, interpretações, e que por meio de uma reflexão crítica as questões norteadoras são clareadas e o projeto é construído. Por final o protótipo é finalizado e os resultados são apresentados.

1.5 PRODUÇÃO CIENTÍFICA

Como partes das pesquisas desenvolvidas, foram publicados os seguintes artigos:

iv. **"Uma Abordagem Arquitetural de um Sistema Pergunta-Resposta"**, publicado na conferência internacional IADIS WWW/INTERNT, 2011.

v. **"Uma Sistema Inteligente Baseado em Ontologia para Apoio ao Esclarecimento de Dúvidas"**, publicado no Simpósio Brasileiro de Informática na Educação, 2011.

1.6 ESTRUTURA DA DISSERTAÇÃO

Este trabalho está organizado da seguinte maneira:

No Capítulo 1, é definido o contexto do projeto de forma introdutória, apresentando uma visão geral. Salientamos a motivação, objetivo, metodologia e a produção científica.

No Capítulo 2, é percorrido um breve histórico sobre as abordagens e sistemas de pergunta-resposta, assim como a estruturação do discurso é detalhada apresentando suas características e definições. São descritos trabalhos que constituem o estado da arte e uma comparação entre os mesmos é feita, na qual o sistema proposto nesta dissertação é enquadrado.

No Capítulo 3, é apresentada uma arquitetura conceitual que é a base fundamental para o desenvolvimento deste trabalho. Detalhamos todos os componentes e as atividades envolvidas no processo de resolução da pergunta e, por final, comparamos as abordagens conceituais da arquitetura com o apresentado no estado da arte.

No Capítulo 4, é descrito de forma geral o funcionamento da arquitetura como um todo. Além disso, detalhamos o funcionamento exclusivo de cada componente tecnológico.

No Capítulo 5, narramos o estudo de caso pela construção de um protótipo, analisamos os resultados e verificamos os objetivos alcançados que foram traçados nas questões norteadoras.

No Capítulo 6, são expostas as considerações finais e as direções futuras para novas pesquisas.

CAPÍTULO 2 SISTEMAS DE PERGUNTA-RESPOSTA

O objetivo deste capítulo é apresentar os conceitos fundamentais e referenciais teóricos que sedimentam esta dissertação. Esse capítulo está organizado da seguinte forma: A Seção 2.1 apresenta o histórico e as evoluções dos sistemas de pergunta-resposta. A Seção 2.2 esclarece as definições e características dessa classe de sistema. A Seção 2.3 apresenta o estado da arte dos sistemas de pergunta-resposta e uma comparação entre os mesmos.

2.1 HISTÓRICO DOS SISTEMAS DE PERGUNTA-RESPOSTA

É possível observar um crescente aumento no interesse na área de sistemas de pergunta-resposta desde a apresentação do tema *Pergunta-Resposta* na Conferência Internacional de Recuperação de Texto, iniciando com TREC-8 em 1999. No entanto, esse interesse recente não é de maneira nenhuma a primeira vez que o assunto tem sido tratado pelos pesquisadores. Na realidade, Simmons (1965) iniciou uma pesquisa intitulada "*Answering English Questions by Computer*" com a afirmação que seu artigo revisava não menos que quinze implementações de sistema de pergunta-resposta para língua inglesa, sistema esses construídos ao longo dos últimos cinco anos. Esses sistemas incluem esquemas de arquitetura de sistema de pergunta-resposta, interface com o repositório de dados estruturados e sistemas que tentam achar respostas para perguntas de fontes textuais, tal como enciclopédia (HIRSCHMAN *et al.*, 2001).

Sistemas de pergunta-resposta têm um legado extenso. Com raízes no método socrático, pergunta-resposta automatizadas marcaram o advento dos computadores com a criação de sistemas para pergunta-resposta a partir de banco de dados como era feito em 1950. (MAYBURY, 2004). O primeiro e mais conhecido programa de pergunta-resposta é o BASEBALL (GREEN *et al.*, 1961), um programa para responder perguntas sobre torneios de baseball jogados na liga americana sobre uma temporada. Dada um pergunta tal como: "De quem *Red Sox* perdeu no dia 5 de julho?" ou "Quantos jogos *Yakees* jogaram em julho?" ou mesmo "Quantos times jogaram em julho?" BASEBALL

analisava a pergunta usando conhecimento linguístico, em forma canônica, e gerava uma consulta em uma base de dados estruturada sobre BASEBALL (HIRSCHMAN *et al.*, 2001).

Esses primeiros sistemas essencialmente se ligavam a um *front end* e *back end* de um sistema de banco de dados. O *front end* realizava análise, interpretação e mapeamento das perguntas redigidas em termos comuns (cotidiano) para formas específicas de serem executadas em um banco de dados – por exemplo: "Qual é a concentração média de alumínio em rochas alcalinas?", seria mapeado para "rochas alcalinas" no banco de dados, localizando a concentração de alumínio em cada, e depois calculando uma média sobre esses valores (CLARK *et al.*, 2010).

Essas primeiras incursões em banco de dados de pergunta-resposta foram essencialmente abandonadas no final de 1980 por duas razões – uma técnica, e outra social. Tecnicamente, um esforço considerável é necessário para garantir uma efetividade e confiança no mapeamento entre perguntas dos usuários e consultas do banco de dados. Não apenas depende de um mapeamento correto para a estrutura de um banco de dados particular, mas muitas perguntas do usuário precisam ser mapeadas para a mesma consulta do banco de dados. Pior ainda, perguntas que diferem minimamente uma das outras precisam ser mapeadas para a mesma consulta do banco de dados. A única solução disponível na época para esses problemas era mapear todas as regras possíveis por um especialista em sistemas. Como uma solução, isso não é escalável e nem portável. O problema social envolve a falta de assistência significativa para a tecnologia: pessoas comuns não tinham acesso às enormes bases de dados e os gestores das bases de dados não tinham interesse em acessar os dados (CLARK *et al.*, 2010). Um resumo dessa história pode ser vista na Figura 2.1.

O primeiro sistema pergunta-resposta baseado na web, MIT'S START, surgiu em 1993 seguido logo por Ask Jeeves em 1996. Ask Jeeves apoiado por consultas em linguagem natural e ambos utilizavam máquinas de buscas na web (MAYBURY, 2004).

Com o advento da Web, este problema social desapareceu, e diversas técnicas se revelam úteis no mapeamento entre perguntas e respostas. Enquanto isso nos dias atuais, não significa que o número crescente de banco de dados contendo informações ricas e

úteis podem voltar a ser preparados para acessos por meio de perguntas em linguagem natural (CLARK *et al.*, 2010).

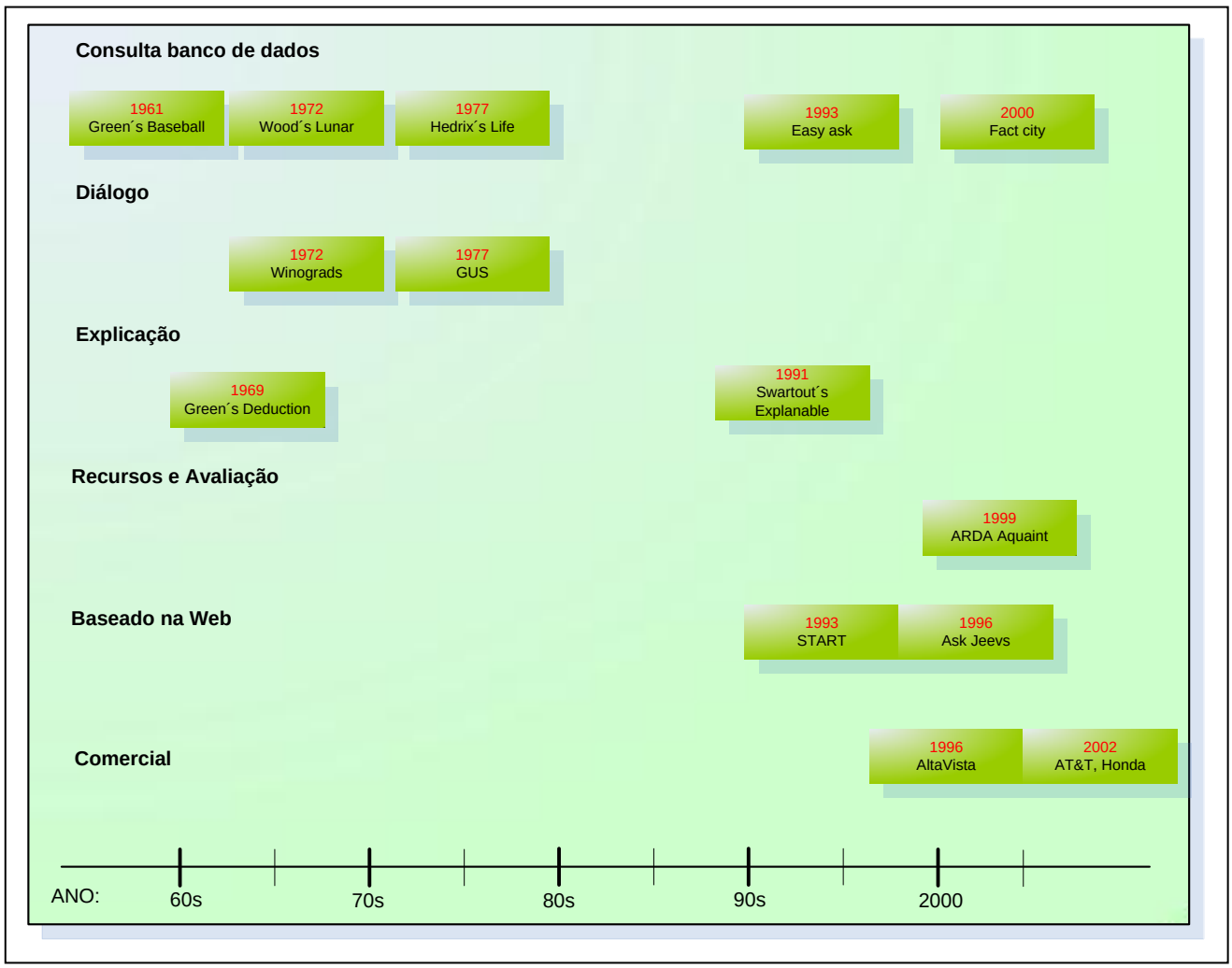

Figura 2.1 – História dos Sistemas Pergunta-Resposta (Fonte: MAYBURY, 2004).

2.2 DEFINIÇÕES E CARACTERÍSTICAS

Para apresentação das terminologias do assunto pesquisado, bem como para clarear o entendimento, detalharemos as características de um sistema de pergunta-resposta.

Em geral, sistemas de pergunta-resposta são baseados em conhecimento adaptados a um domínio particular e com interfaces em linguagem natural. (AKERKAR *et al.*, 2010). A característica “baseados em conhecimento” armazena muitas informações importantes, que será explicado no extrato a seguir:

> *"Sistemas baseados em conhecimento são sistemas que usam inteligência artificial para resolver problemas. Incorpora um banco de dados de conhecimento do perito com utilidades para facilitar o conhecimento recuperado em resposta para uma consulta específica, juntamente com aprendizagem e justificação, ou para transferência da experiência de um domínio de conhecimento para outro. Em particular, sistemas baseado em conhecimento focam no uso de técnicas para apoiar a tomada de decisão humana, aprendizagem e ação. Tais sistemas são capazes de cooperar com humanos e são usados para resolução de problema, treino e auxílio ao usuário e especialista do domínio." (AKERKAR et al., 2010).*

A área de pesquisa de um sistema de pergunta-resposta é um desafio, em parte por causa da intersecção de muitos campos científicos incluindo processamento de linguagem natural (entendendo e gerando textos em linguagem natural), recuperação de informação (formulação das consultas, análise dos documentos, retorno dos documentos relevantes), e interação humano-computador (projeto de interface, modelagem.). Muitas disciplinas adicionais podem apoiar um sistema de pergunta-resposta e que não são mostradas na Figura 2.2, por exemplo: representação do conhecimento e raciocínio para perguntas e análise da resposta, ou algoritmos para achar respostas de preferência, ou extração em fontes de áudio ou vídeo, entre outras (MAYBURY, 2004).

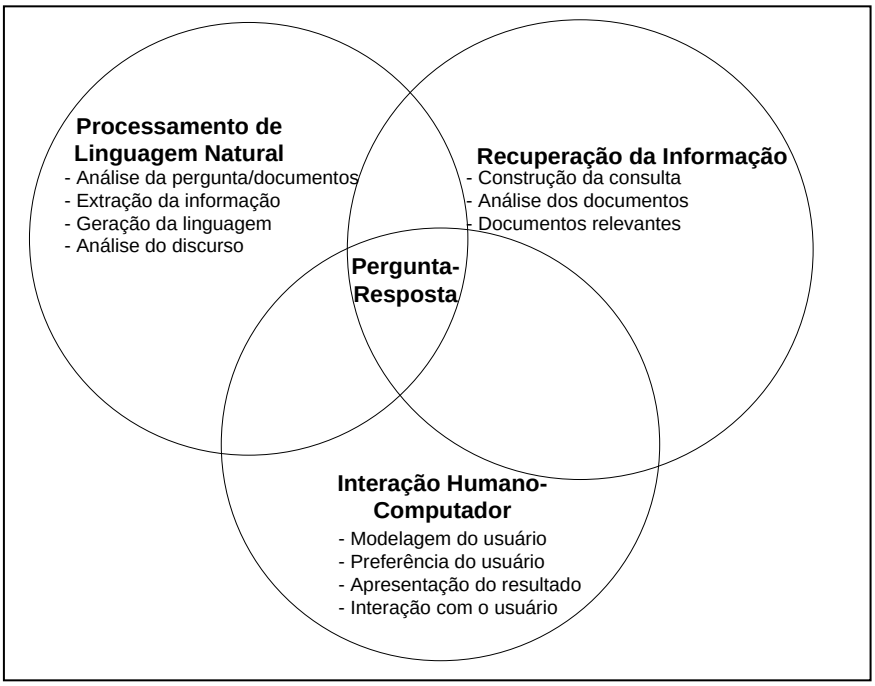

Figura 2.2 – Pergunta-resposta e outras disciplinas (Fonte: MAYBURY, 2004).

Em torno das diferentes disciplinas e áreas de pesquisas envolvidas, surgem várias classificações. A seguinte seção apresenta algumas perspectivas e classificações para os sistemas de pergunta-resposta abordados na literatura.

2.2.1 Classificação dos Sistemas Pergunta-Resposta

A criação de um esquema de classificação para um sistema de pergunta-resposta é uma tarefa bastante complicada. A dificuldade reside principalmente na seleção da perspectiva em que essas classificações devem ser cumpridas e na grande variedade de abordagens. A seguir apresentamos classificações que levam em conta diferentes perspectivas.

A taxonomia apresentada por Moldovan *et al.* (1999) propõem a classificação dos sistemas de pergunta-resposta dividida em 5 classes, dependendo de 3 características principais: (a) o nível de conhecimento necessário, (b) o nível de raciocínio, e (c) a indexação e técnicas de linguagem natural utilizadas.

As bases de conhecimento e sistemas de raciocínio fornecem meios para facilitar a construção do contexto da pergunta e para encontrar a resposta na coleção de documentos. Por outro lado, as técnicas de indexação permitem que os sistemas

localizem os extratos dos documentos que são prováveis possuidores da resposta. Finalmente, técnicas de processamento de linguagem permitem uma precisa localização e extração da resposta (VICEDO *et al.*, 2007).

Moldovan *et al.* (2003) caracterizou os sistemas de acordo com a complexidade das perguntas e a dificuldade de extração. A Tabela 2.2 organiza as classes descritas por Moldovan.

Tabela 2.1 – Caracterização dos sistemas por Moldovan.

Classe 1: capaz de processamento de perguntas factuais	Estes sistemas extraem a resposta como trecho de texto de um ou mais documentos. Frequentemente as respostas são encontradas literalmente em textos ou como uma variação morfológica. Tipicamente, a resposta é extraída usando métodos empíricos que se apoiam na manipulação de palavras-chaves.
Classe 2: capaz de mecanismos de raciocínio	Nesta classe, as respostas são encontradas em trechos de texto, mas ao contrário da classe 1, a inferência é necessária para relacionar a pergunta a resposta. Além de elaborar métodos de detecção de resposta tais como ontologias ou codificação do conhecimento pragmático. Alternativas semânticas, conhecimento axiomatizado, e métodos de raciocínio simples são necessários. Um exemplo seria a pergunta: "Quando Sócrates morreu?", no qual "morreu" deve ser ligado com o ato de beber vinho envenenado. A *Wordnet* e suas extensões são às vezes usadas como fontes de conhecimentos gerais.
Classe 3: capaz de fusão de resposta de diferentes documentos	Nesta classe, respostas parciais estão espalhadas em vários documentos e as fusões das respostas são necessárias. A complexidade aqui varia pela composição de listas simples até difíceis operações como perguntas que estruturam um plano de ação, (ex: Como eu monto uma bicicleta?)
Classe 4: sistemas interativos	Estes sistemas são capazes de responder perguntas no contexto de interações anteriores com o usuário. Como relatado em Harabagiu *et al.* (2001), processando uma lista de perguntas feitas em um contexto envolve uma resolução de referência complexo. Ao contrário resolução comum, algoritmos verificam anáforas a partir de perguntas atuais ou de perguntas anteriores, ou de uma resposta anterior.

Classe 5: capaz de raciocínio por analogia	As características destes sistemas são suas habilidades para especular perguntas, por exemplo: "Será que o banco vai elevar a taxa de juros na próxima reunião?" "A indústria de aviação está com problemas?" Uma vez que as respostas para as perguntas não são não explicitamente indicadas em documentos, simplesmente porque eventos podem ainda não ter acontecido, sistema de pergunta-resposta a partir desta classe deve decompor as perguntas em consultas para extrair pedaços de evidencias, depois a resposta é formulada usando raciocínio por analogia.

Vicedo *et al.* (2007) apresentaram uma classificação de acordo com o nível da análise da linguagem natural. Essa taxonomia propôs as três seguintes classes:

- Classe 0: Não usam técnicas processamento de linguagem natural – Os sistemas nesta classe aplicam técnicas tradicionais de recuperação de informação que foram adaptadas para tarefa de pergunta-resposta. Eles recuperam pedaços pequenos que acreditam conter a resposta esperada. A análise da pergunta geralmente consiste em selecionar aqueles termos da pergunta que devem aparecer na resposta. Para este propósito, as *stopwords*[6] são eliminadas e termos com alto valor de discriminação são selecionados.
- Classe 1: Usam técnica superficial de processamento de linguagem natural – Este tipo de abordagem realiza uma análise mais detalhada da pergunta, a qual permite mais precisão na identificação das respostas. De forma geral, o processo de analisar a perguntas opera o processo de identificar o tipo semântico da entidade necessária como resposta (uma data, um nome de uma pessoa, uma localização, um número etc.) e restringe e relaciona características para o tipo esperado para a resposta identificada. Isso pode incluir palavras-chaves da pergunta que permitiram recuperar aqueles textos que provavelmente conterão a resposta, identificando qualquer relacionamento sintático e/ou semântico entre as entidades da pergunta e a resposta candidata.
- Classe 2: Usam técnicas complexas de processamento de linguagem natural – técnicas complexas de processamento de linguagem natural raramente são usada

[6] *Stopwords* – são palavras que não carregam significado e, portanto podem ignoradas. (BAEZA-YATES *et al*., 2000)

> em pergunta-resposta, principalmente devido as dificuldades intrínsecas relacionadas a representação do conhecimento. Em geral, esses sistemas representam perguntas e respostas candidatas por meio de lógica formal.

(AKERKAR *et al.*, 2010) apresenta uma taxonomia para sistema pergunta-resposta baseada nos seguintes critérios:

- Recursos linguísticos e do conhecimento.
- Envolvendo processamento de linguagem natural.
- Processamento de documento.
- Métodos de raciocínio.
- Hipótese sobre resposta que são explicitamente indicadas nos documentos
- Necessidade de gerar respostas.

Maybury (2004) apresenta uma categorização baseada na especialidade da extração e retorno da resposta:

- Sistema de pergunta-resposta temporal – A interpretação automatizada de perguntas com elementos temporais tais como tempo relativo ou absoluto, ponto (momento exato), duração, e extração da resposta com aspectos temporais.
- Sistema de pergunta-resposta espacial – Fornecem resposta envolvendo objetos espaciais (exemplo: Localização, regiões), atributos (exemplo: tamanho, forma), e relações (exemplo: acima/abaixo, dentro/fora, próximo/longe), possivelmente precisam de inferência espacial.
- Sistema de pergunta-resposta definível – A criação automatizada de definição ou descrição dos objetos.
- Sistema de pergunta-resposta biográfico – A criação automatizada da resposta relacionada a perguntas sobre características e eventos na vida de uma pessoa, grupo ou organização.
- Sistema de pergunta-resposta que emite parecer – A detecção automatizada de opiniões (de indivíduos, grupos, ou instituições), e responde a perguntas sobre ponto de vista e perspectivas relacionado a conteúdo subjetivo.

- Sistema de pergunta-resposta multimídia – A resposta pode ser expressa por um vasto conteúdo dinâmico ou estático (exemplo: texto, áudio, imagens, vídeo, entre outras).
- Sistema pergunta-resposta multi-idiomas – Respondendo a perguntas tanto para usuários com diferentes idiomas ou de fontes multi-idiomas que exigem: tradução na recuperação, extração do conteúdo em uma língua estrangeira e geração da resposta em linguagem específica.

Zheng (2002) apresentou uma classificação para automatização de pergunta-resposta baseando em domínio fechado (ou específico) e domínio aberto. O domínio fechado seria baseado em bases de conhecimento ou corpus pré-análisados. O domínio aberto tenta construir um sistema de pergunta-resposta com vastas coleções de documentos na web. No artigo apresentado por Bouma *et al.* (2011) é realizada uma comparação por meio da exemplificação com o domínio médico: "Para o domínio fechado, exemplo médico, o número de tipos de perguntas é limitado. A maioria das perguntas são sobre: definição, causas, sintomas, e tratamentos. Isto sugere que a extração pode ser muito efetiva para um sistema médico. Um problema de um domínio específico é fato que o *corpus* tende a ser menor que aquele usado em um domínio aberto, e assim existe uma frequência bem menor das instâncias (entidades) procuradas. Em segundo lugar, enquanto que no domínio aberto a extração se concentra na aprendizagem de entidades, e sabemos que as entidades das relações médicas são frequentemente substantivos em frases complexas, que estão sujeitas a uma grande variação gramatical. Isso é um fator adicional que reduz bastante a identificação de instâncias. Portanto a maioria dos sistemas do domínio médico, tem feito uso das duas abordagens para tornar a tarefa viável. Primeiro, o uso de um tesauros para identificar os conceitos do domínio. Segundo, ao invés de aprendizagem dos termos, extração por meio de padrões em *corpus* anotados."

2.2.2 A Arquitetura de um Sistema de Pergunta-Resposta

Como ilustrado na Figura 2.3, sistemas de pergunta-resposta tipicamente são dirigidos para: a) alguma série de perguntas (exemplo: *WH-question*: Quem, Qual, Quando, Como, Porque) b) processar uma variedade de fontes (exemplo: documentos, páginas web, banco de dados), a fim de produzir respostas para os usuários. Um sistema de

pergunta-resposta contém um módulo ou componente para analisar perguntas, recuperar fontes, extração da resposta, e apresentação da resposta, com a possibilidade de aprovação do usuário para melhorar o processo com o tempo. No futuro esperamos que os sistemas possam ser melhores, sendo genéricos para um usuário qualquer ou adaptados a um usuário específico, fornecendo as melhores ou todas as respostas, fornecendo respostas junto com suas justificações ou explicações, e apresentando uma gama selecionável de resposta, tais como entidades (exemplo: pessoa, localização, ou tempo para perguntas do tipo Quem, Onde e Quando), fragmentos ou passagens de fontes de respostas, ou fontes completas junto com a resposta (MAYBURY, 2004). Nesta seção, baseado na visão de Clark *et al.* (2010), Athenikos *et al.* (2010), e Hirschman *et al.* (2001) apresentaremos os principais componentes de uma arquitetura de pergunta-resposta. A Figura 2.3 ilustra a arquitetura genérica.

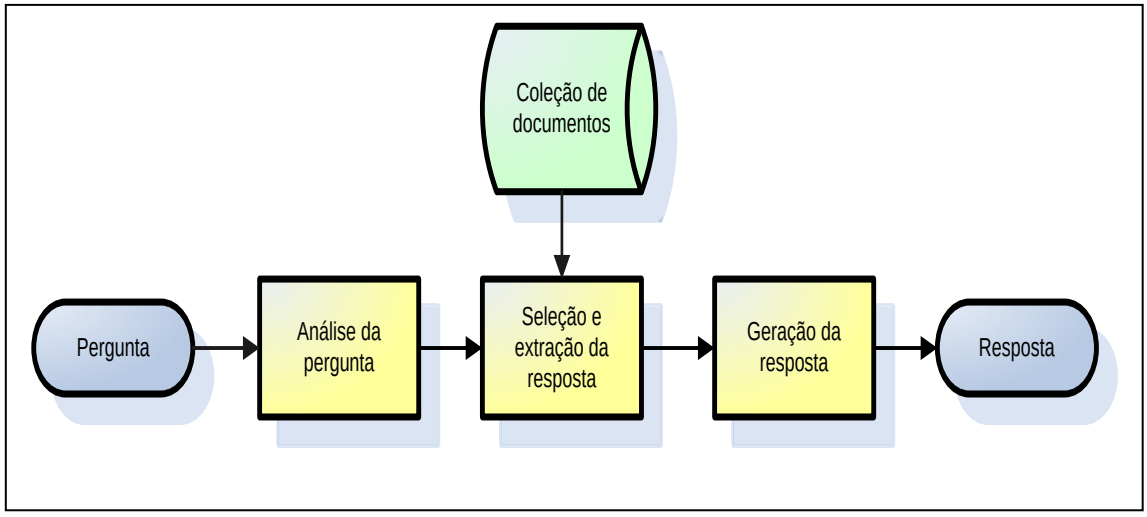

Figura 2.3 – Arquitetura básica de um sistema de pergunta-resposta (Fonte: MAYBURY, 2004).

Para satisfazer necessidades funcionais de uma arquitetura de um sistema de pergunta-resposta citamos alguns dos seguintes pontos propostos por (MAYBURY, 2004):

- Modularidade – O sistema deve encapsular e separar (definir bem os limites) os módulos (análise da pergunta, recuperação, extração, formulação da resposta, seleção da resposta, entre outras.), e os módulos devem permitir integração para apoiar avaliação comparativa, reuso, entre outros.
- Múltiplo/Flexível fluxo de dados – O sistema deve apoiar uma variedade de interconexões entre os componentes. Se necessário o sistema deve distribuir o trabalho para múltiplos agentes.

- Fontes de dados heterogêneas – O sistema deve incorporar abstrações apropriadas para apoiar acesso direto a fontes de dados com múltiplos domínios e categorias.
- Apoio para raciocínio inferencial – O sistema deve produzir respostas que não podem ser encontradas em pesquisas diretas, mas deve utilizar inferências a partir das fontes de dados.

De forma geral o processamento de um sistema pergunta-resposta consiste em três fases principais: a análise da pergunta, seleção e extração da resposta e a geração da resposta (ATHENIKOS *et al.*, 2010). Nas próximas seções detalharemos o mecanismo de funcionamento de cada um dos três componentes da arquitetura.

2.2.2.1 Análise da Pergunta

A análise da pergunta está intimamente relacionada com o campo de pesquisa denominado expansão automática da consulta (AQE – *Automatic Query Expansion*). A relativa ineficácia dos sistemas de recuperação de informação é, em grande parte causada pela imprecisão com que uma consulta formada por algumas palavras chaves modela a informação que o usuário necessita. Para melhorar o estágio de recuperação dos sistemas pergunta-resposta, uma estratégia é expandir a pergunta original com termos que são esperados que apareçam nos documentos. Diferentes abordagens AQE são aplicadas, usando ontologias, *Wordnet*, análise sintática, análise semântica de perguntas baseada em regras, estatística, entre outras (CARPINETO *et al.*, 2012) (CHIRITA *et al.*, 2007).

Muitos sistemas usam na etapa de análise da pergunta módulos para reconhecimento do tipo da pergunta baseado na estrutura sintática e no tipo semântico da resposta esperada (PRAGER *et al.*, 2000), frequentemente utilizando *Wordnet*. A análise da pergunta é usada para dois propósitos: palavras-chaves são extraídas e usadas como termos de uma consulta para recuperar documentos candidatos de uma coleção de documentos. O outro propósito, é utilizar as análises para a fase da extração da resposta, ou seja, classificação da pergunta a fim de obter a resposta. Muitos sistemas não apenas extraem palavaras chaves na análise da pergunta para utilizar como termos de uma consulta, mas além disso modificam a consulta conforme o tipo da pergunta (MONZ, 2003). Isso pode ser

feito adicionando termos suplementares à consulta. Por exemplo: adicionando os termos como “metro” ou “kilometro” para a pergunta “Qual o tamanho...?”, irá assegurar que apenas documentos que contenham alguma medida sejam encontradas (FLIEDNER, 2007).

Nesta fase de análise da pergunta geralmente são necessários dois processos para identificar qual o tipo da informação está sendo procurado (classificação da pergunta) e em qual pedaço de texto provavelmente pode ser encontrado (construção da consulta). Embora esse processo possa ser realizado em paralelo, a construção da consulta está intimamente ligada com a recuperação do texto que será discutido na próxima seção. (CLARK *et al.*, 2010). A classificação da pergunta visa associar uma etiqueta, indicando o tipo da informação a ser procurada – por exemplo, o significado de uma abreviação (ABREV) ou de alguma palavra ou frase (DEFINIÇÃO), o nome de pessoas que têm ou tinham uma propriedade particular ou conjunto de propriedades (PESSOA) etc. Essa etiqueta fornece restrições semânticas testáveis sobre as respostas candidatas possíveis. As etiquetas atribuídas para pergunta têm sido usada para apoiar a recuperação de texto por meio de anotação preditiva (PRAGER *et al.*, 2000), assim como para apoiar o processo envolvido com a identificação e ordenação das perguntas candidatas. Podem ser construídas regras para classificação da pergunta, exemplo: Se a pergunta iniciar com Quem ou De quem, a classificação da pergunta é PESSOA. Se a pergunta iniciar com Onde , a classificação é LOCALIZAÇÃO. Alguns outros sistemas (LI *et al.*, 2002; MOSCHITTI *et al.*, 2007) usam classificadores probabilísticos, onde uma pergunta Q é representada por um conjunto de características (CLARK *et al.*, 2010).

2.2.2.2 Seleção e Extração da Resposta

A fase de seleção e extração da resposta preocupa-se em recuperar os documentos apropriados e extrair um conjunto de respostas candidatas. Existem duas formas de recuperar a resposta: recuperação baseada em relevância e recuperação baseada em padrão.

Na recuperação baseada em relevância as consultas são interpretadas como pedidos de textos relevantes para um determinado assunto. A relevância pode ser avaliada por meio

de uma combinação booleana dos termos, vetor de termos ponderado ou um modelo de linguagem, assim como na recuperação padrão de textos. O problema é que um sistema de pergunta-resposta demanda respostas, ao invés de textos que as contêm. Embora a indexação baseada em palavras dos textos seje rápida e eficiente, essa indexação caracteriza textos em termos de suas propriedades léxicas, não pela informação local. Por exemplo, um jornal em 1987 anunciou um desastre com uma balsa mencionando a seguinte passagem: “Foi o pior desastre em tempo de paz envolvendo um navio britânico desde a época que o Titanic afundou em 1912.” Enquanto essa frase contém a resposta para a pergunta: Quando Titanic afundou? Essa informação local pode simplesmente ser um ruído com relação às propriedades léxicas mais proeminentes do texto. Dessa forma pode não ser recuperado na busca baseada por relevância por “Titanic” e “afundou”, ou pode não ter uma alta pontuação em um processo de classificação suficiente para a extração de respostas candidatas. E assim, nunca chegar à resposta. Uma solução largamente adotada para o problema de localidade na recuperação baseada em relevância é simplesmente quebrar o texto em conjuntos de passagens separadas, cada qual indexado separadamente para a recuperação do texto, assumindo que existe uma melhor correlação (CLARK *et al.*, 2010).

Outra técnica de recuperação de texto usada é a anotação preditiva (PRAGER *et al.*, 2000). Os textos são indexados não apenas pela posição de cada palavra no texto, mas também pela posição de cada um dos 20 tipos de entidades, cada qual poderia responder uma pergunta de um tipo determinado. Por exemplo, a anotação preditiva da frase: Sri Lanka tem maior renda per capita do que a África do Sul. O índice seria a entidade COUNTRY$ como ocorrência sobre as duas primeiras palavras da frase, e PLACE$ como ocorrência sobre as duas últimas. A notação preditiva pode ser usada como parte da construção da consulta para recuperação da pergunta. Quando a pergunta é mapeada para a consulta, não apenas as palavras-chaves são incluídas na consulta, mas também entidades apropriadas para o tipo de pergunta, por exemplo, (uma pergunta “Onde”): Onde é a capital do Sri Lanka? As palavras chaves “capital”, “Sri” e “Lanka” seriam mapeadas para as entidades (LUGAR$, PAIS$, ESTADO$, NOME$), todos podem potencialmente responder a pergunta “ONDE” (CLARK *et al.*, 2010).

A recuperação baseada em padrão difere da recuperação baseada em relevância no fato de retornar um pedaço como evidência de uma correspondência (combinação) ao invés

de ser um ponteiro para o texto que contém o trecho da resposta. Assim a recuperação baseada em padrão não obtém qualquer benefício de textos quebrados em pequenos pedaços, como é feito na recuperação baseados em relevância, porque visa os trechos ao invés das fontes. Padrões geralmente refletem relacionamentos diretos entre a pergunta e sua resposta. Por exemplo, (1) Quando foi inventado o telefone? Um padrão de resposta é encontrado na pergunta: O telefone foi inventado em <resposta>. Outros relacionamentos sintáticos predizem outros padrões de resposta, exemplo: (1) Inventaram o telefone em <resposta>, (2) O telefone, inventado em <resposta>, (3) em <resposta> o telefone foi inventado. Outras pesquisas mostram como utilizar outras fontes léxicas para melhorar essa recuperação, tal como *Wordnet* e *FrameNet*[7] (CLARK *et al.*, 2010).

Existem outras inúmeras técnicas para recuperação da resposta que podem ser encontradas em Tellex *et al.* (2003) e Grappy *et al.* (2011) que utilizam implicação textual (DAGAN *et al.*, 2009) (RTE – *Recognizing Textual Entailment*) como tarefa para verificar a resposta. O RTE é uma tarefa de determinação se um dado pedaço de texto T implica em outro pedaço de texto H. Assim um trecho de texto candidato a resposta pode ser deduzido de uma pergunta. O assunto RTE será detalhado no Capítulo 4. Torres-Moreno (2009) utiliza sumarização de textos, algoritmos e técnicas estatísticas para extrair a resposta.

2.2.2.3 Geração da Resposta

O módulo de geração da resposta geralmente manipula textos. Entretanto existem alguns sistemas na literatura que utilizam outras formas de expor a resposta, exemplo: som e imagem (BOSCH *et al.*, 2011), vídeos (LEI *et al.*, 2010), orientada por serviços (WANG *et al.*, 2012), múltiplos idiomas (FERRÁNDEZ *et al.*, 2011), iniciativas de diálogos para decidir como proceder (KIYOTA *et al.*, 2002). Alguns outros sistemas utilizam de um *feedback* (JURCZYK *et al.*, 2007) para melhorar a recuperação da respostas. São diversas as possibilidades para apresentar a respostas e melhorar a interação com o usuário.

---

[7] https://framenet.icsi.berkeley.edu/fndrupal/framenet_data

2.2.3 A Pergunta

Experiências têm demonstrado que não é fácil determinar quais características tornam algumas perguntas mais difíceis de responder que outras. Portanto, para fins do sistema de pergunta-resposta, perguntas são geralmente classificadas dependendo do tipo da resposta necessária para pergunta (VICEDO *et al.*, 2007).

As perguntas podem ser analisadas a fim de obter informação da resposta esperada. Uma categoria sintática é atribuída para a pergunta, dependendo da forma sintática da pergunta. Entretanto, o mesmo tipo da pergunta pode ser expresso em diferentes formas sintáticas. Por exemplo, as seguintes perguntas podem combinar um pedido de localização (STRZALKOWSKI, 2008):

- Pergunta 1: Qual é a localização no EUA da Procter&Gamble ?
- Pergunta 2: Onde está a Protect&Gamble nos EUA?

Categorização da resposta de acordo com o tipo da resposta é, portanto útil para conhecer qual o padrão da resposta será extraída. Padrões transmitem a relação que suportam uma categoria, e assim são específicos para esta categoria. Infelizmente, essa não é uma verdade para todos os padrões. Alguns podem pertencer a mais de uma categoria (STRZALKOWSKI, 2008).

Vicedo *et al.* (2007) classificaram as perguntas em:

- Pergunta factual (produzem fatos): Estas perguntas requerem uma ou mais itens específicos de dados, por exemplo: data, localidades, quantidade, entre outros – por exemplo, Qual é a capital do Brasil? Quando Bob Marley morreu?
- Pergunta de síntese: Estas perguntas exigem que o sistema localize instâncias específicas de informação e as resuma para apresentação. Exemplo: Quais são as três residências da rainha Elisabeth II do Reino Unido? Quais as questões que George Bush tratou em sua última visita na Alemanha?
- Pergunta de contexto: Estas perguntas são postadas em um contexto de perguntas anteriormente processadas. Dessa forma, a interpretação da pergunta dependerá do significado de uma ou mais respostas anteriores. Exemplo: (1)

Qual país foi o primeiro a usar telescópio com óptica adaptativa? (2) Onde o país está localizado? (3) Qual é o nome do maior laboratório localizado lá?

- Perguntas especulativas: Estas são perguntas complexas que precisam utilizar técnicas dedutivas – por exemplo: O que poderia acontecer em Marrocos se o Rei Hassan II fosse assassinado?

Para aprimorar na extração da resposta (KONCHADY, 2008) apresenta uma lista exemplificando a classificação das perguntas (Tabela 2.3).

Tabela 2.2 – Classificação das perguntas por Konchady (2008).

Pergunta	Tipo	Entidade/Processo
Quem é Alvaro Uribe Velez?	Fato	Pessoa
Qual planta pode ser usada no tratamento de pressão sanguínea?	Fato	Coisa
Qual é a maior cidade em Myanmar?	Fato	Lugar
Khan visitou a Coreia do Norte em dezembro de 1994?	Procedimental	Raciocínio
O Iran tem mísseis que podem alcançar Tel Aviv?	Procedimental	Raciocínio
Qual é a altura de Christina Aguilar?	Fato	Dimensão
Onde o musgo cresce?	Fato	Localização
O que é um ruído?	Fato	Definição
Como eu consigo remover ferrugem?	Procedimental	Tutorial

De acordo com (KONCHADY, 2008) as respostas do tipo fato são frases curtas ou uma única palavra. As respostas procedimentais podem consistir de explicações ou frases longas.

2.3 ESTADO DA ARTE

Esta pesquisa bibliográfica faz uma explanação detalhada dos sistemas de pergunta-resposta mais atuais. Para permitir comparação o sistema deve basear seus testes na métrica: “percentual de questões respondidas corretamente”. No entanto, esta pesquisa reúne apenas os trabalhos livres para *download*.

O FREyA (DAMIJANOVIC *et al.* 2010) traduz uma pergunta em linguagem natural ou palavras chaves em uma consulta SPARQL[8], e retorna a resposta para o usuário depois de executar uma consulta na ontologia. A dinâmica do sistema pode ser resumida nos seguintes passos: Identifica e verifica os conceitos da ontologia, gera a consulta SPARQL e identifica o tipo da resposta e apresenta o resultado para o usuário. O algoritmo para traduzir uma pergunta em linguagem natural em um conjunto de conceitos da ontologia combina análise sintática com raciocínio na ontologia. Nos casos em que o algoritmo não infere conclusões automaticamente, sugestões são geradas para o usuário. Ao envolver o usuário em um diálogo, têm-se melhores chances de identificar as informações que são consideradas ambíguas. Na fase de identificação dos conceitos, é utilizado conhecimento disponível na ontologia para reconhecer e anotar na pergunta com os termos da ontologia. Se existir anotações ambíguas na consulta, é realizado com o usuário um diálogo. A próxima fase identifica os conceitos potenciais da ontologia que são derivados de uma análise sintática. Por exemplo, cada substantivo é identificado como um conceito potencial. O passo seguinte é mapear os conceitos potenciais aos conceitos da ontologia, que pode ser feito de duas formas: automaticamente ou pela interação com o usuário. Esse sistema também utiliza algoritmos de similaridades de *string* e os sinônimos da *Wordnet* para auxiliar na seleção dos conceitos potenciais. Depois de resolver todos os conceitos potenciais, é gerado um conjunto de triplas com os conceitos que serão combinados com elementos da sintaxe da consulta SPARQL a fim de gerar uma consulta. O resultado da consulta é um grafo. Os testes obtiveram um recall de 92.4% sobre um total de 250 questões.

O trabalho apresentado por Oh *et al.* (2012) esboça uma arquitetura dividida nos seguintes módulos básicos: extração da resposta, análise da pergunta e geração da resposta. O módulo *análise da pergunta* recebe uma pergunta em linguagem natural e realiza um processamento empregando várias técnicas de análise linguística (ex: *tagging, chuncking,* entre outras) e algumas de análise semântica. Por meios desses processamentos e análises é gerada a representação: Pergunta = {AF, AT, QT, AS}, onde AF, AT, QT, AS simbolizam: o formato da reposta, o assunto da resposta, o objetivo da resposta e a fonte esperada da resposta. Por exemplo, a pergunta "Onde Mozart nasceu?" é enviada pelo módulo *análise da pergunta* que determina o formato

[8] SPARQL é uma linguagem de consulta para RDF. http://www.w3.org/TR/rdf-sparql-query/

da resposta como “único”, o assunto da resposta “data”, o objetivo da resposta “Mozart” e “lugar nascimento”, e a fonte da resposta é a base de conhecimento de pergunta-resposta gerais. A resposta é extraída da base com um valor de confiança associado. Se o valor de confiança for menor que um valor limite aceitável ou a resposta não for encontrada, o módulo de recuperação de texto é acionado. O fluxo geral utiliza um algoritmo de aprendizagem para auxiliar a seleção da resposta. Das cinco estratégias de teste, o melhor resultado obtido (*Automatic strategy-drive*) foi realizado com 500 questões e apresentou 84% de respostas corretas.

No trabalho de Liu *et al.* (2010) é proposto um tipo de método de recuperação de pergunta-resposta baseado em Perguntas Frequentes (FAQ - *Frequently Asked Questions*). Geralmente sistemas de pergunta-resposta baseado em FAQ, combinam a pergunta do usuário com as do banco de dados de pergunta-resposta, e retornam as respostas ao usuário. O sistema consiste basicamente de três partes: interpretação da pergunta, recuperação da informação e gerenciamento do banco de dados de FAQ (atualizar e inclusão). Antes de realizar a inclusão no banco de FAQ, a pergunta será tratada. O processo inclui: análise da pergunta, determinação do padrão de consulta, anotação da pergunta, resumo e indexação da pergunta anotada. O módulo de interpretação da pergunta realiza um cálculo de similaridade entre as palavras-chaves de uma pergunta anotada por uma ontologia e um conjunto de palavras chaves padrão do banco do FAQ. O cálculo de similaridade é baseado na técnica matemática do modelo de espaço vetorial. O primeiro passo do módulo da extração da pergunta é gerar um consulta SPARQL da pergunta, que irá recuperar da ontologia a resposta. O teste apresentou um percentual médio das respostas corretas de 72.1%.

O *PowerAqua* (LOPEZ *et al.*, 2011) é uma evolução de outro sistema chamado *Aqualog*, um sistema baseado em ontologia. Na arquitetura do *PowerAqua*, o componente análise da questão utiliza um componente linguístico para processar a consulta. A saída desse componente é um conjunto de triplas linguísticas (< sujeito, predicado, objeto>) que é mapeado para a consulta do usuário. Assim é possível realizar

buscas das respostas em bases OWL/RDF[9]. Os resultados obtidos nos testes apresentaram 48 (69,5%) questões respondidas das 69 questões totais.

O sistema de pergunta-resposta apresentado por Konopík *et al.* (2010) é especialmente efetivo em respostas para questões "*Wh*" (O que, Quem , Quando Onde, Por que, De quem, Qual e Como) sobre pessoas, datas, nomes e localizações. A resposta é construída a partir de dados colhidos na internet, ontologias públicas e conhecimento da linguagem Tcheca. Na apresentação o usuário avalia se a resposta é correta ou incorreta. Essa avaliação é armazenada e utilizada para otimizar o sistema. O teste foi realizado com um conjunto de 100 questões e apresentou 64% de respostas corretas.

Estudando trabalhos na literatura que buscam resolver o problema de resolver uma pergunta e devolver uma resposta exata, encontramos tendências que estão sintetizadas na Tabela 2.3.

Tabela 2.3 – Estado da arte dos sistemas pergunta-resposta.

Referência	**Análise da Pergunta**	**Seleção e Extração**	**Geração das Respostas**
(DAMIJANOVIC *et al.*, 2010)	*Parser*, raciocínio na ontologia, algoritmo que mede similaridade entre string, identifica o tipo da questão.	Gera e realiza consulta SPARQL, aprendizagem de máquina semi-supervisionada.	Web/textual
(OH *et al.*, 2012)	Técnicas de análise linguística (*POS tagging, chunking, named entity tagging*), análise semântica etc.	Aprendizagem de máquina e algoritmo baseado em peso.	Textual
(LIU *et al.*, 2010)	Remoção de *stopwords*, determinação de palavras chaves, ontologias etc.	Gera consulta SPARQL, cálculo de similaridade baseado em métodos estatísticos e semânticos.	Textual

[9] OWL/RDF são linguagens para construção de ontologias.

Referência	**Análise da Pergunta**	**Seleção e Extração**	**Geração das Respostas**
(LOPEZ *et al.*, 2011)	Análise sintática, expressões regulares, *Wordnet*, algoritmo que mede similaridade entre string etc.	Consulta SPARQL, algoritmo para classificar a resposta etc.	Web/textual
(KONOPÍK *et al.*, 2010)	Lematização, *Wordnet*, classificação e extração das entidades, *POS Tagging* etc.	Processamento estatístico	Textual

CAPÍTULO 3 MODELO CONCEITUAL

Este capítulo apresenta uma arquitetura conceitual para um sistema de pergunta-resposta. Essa arquitetura estende-se por todo trabalho como modelo para o desenvolvimento do projeto e apresentação dos resultados.

Iniciamos na Seção 3.1 abordando a visão geral da arquitetura, tais como seus componentes e objetivos principais.

Em seguida, na Seção 3.2, descrevemos as atividades da arquitetura para que a pergunta seja resolvida. Cada uma das atividades é agrupada em um componente da arquitetura específica por desempenhar funcionalidades bem delimitadas, que apresentaremos nas subseções: 3.2.1 Análise da Pergunta, 3.2.2 Seleção e Extração da Pergunta, 3.2.3 Base de Conhecimento e 3.3.4 Geração da Resposta.

Na Seção 3.3 comparamos os conceitos da arquitetura proposta neste trabalho com a pesquisada no estado da arte, mostrando em que aperfeiçoamos e quais são as principais diferenças.

3.1 VISÃO GERAL DA ABORDAGEM

Em engenharia de software, a distinção entre especificação funcional e implementação de sistema é frequentemente discutida como uma separação de “O que” e “Como”. Na especificação das fases, “O que” o sistema deve fazer é estabelecido na interação com os usuários. “Como” está relacionado com as funcionalidades do sistema durante o projeto e implementação (exemplo: quais algoritmos podem ser aplicados). (FENSEL, 2000). Nesta seção ilustramos de forma geral “O que” o sistema deve fazer para responder a pergunta do usuário.

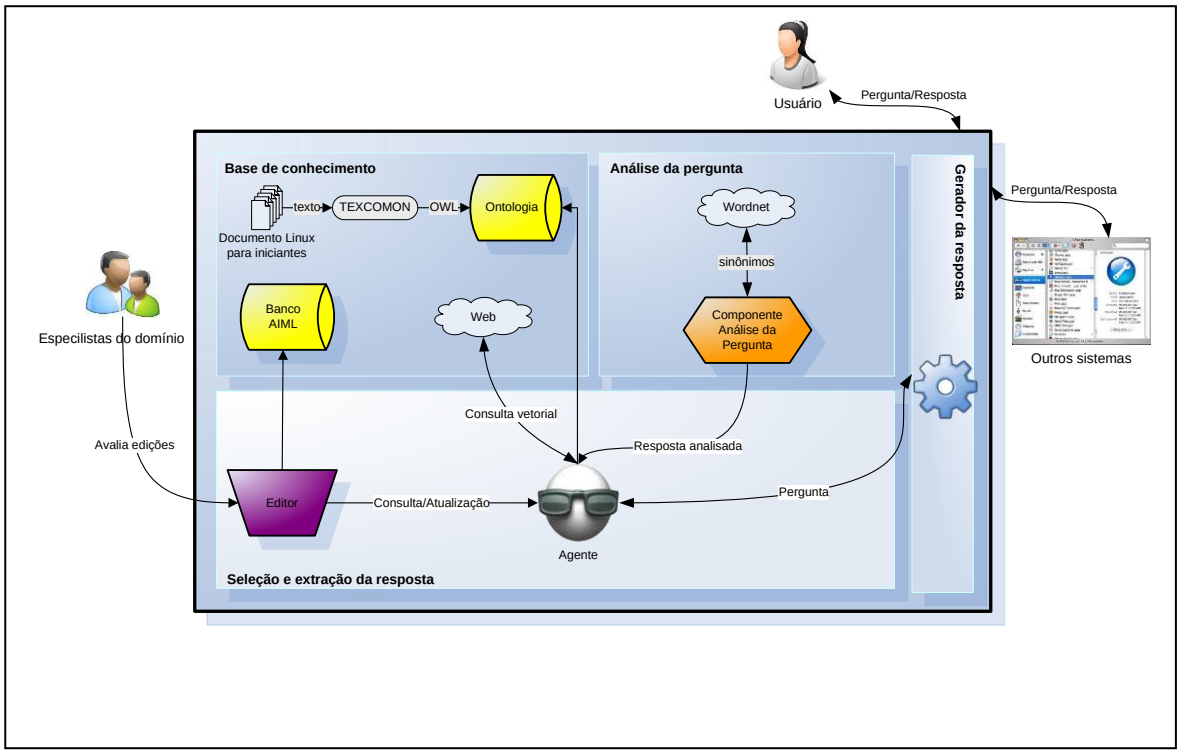

Figura 3.1 – Arquitetura do sistema de pergunta-resposta proposto.

A especialidade da arquitetura descrita neste trabalho (Figura 3.1) resume-se em resolver perguntas da língua inglesa do tipo *WH-Question*, que são: O que, Quem, Quando, Onde, Quais, Quem. Esses tipos de pergunta requerem respostas descritivas ou compostas por fatos (data, local, etc.). Nesse caso optamos por excluir a resolução de perguntas que exigem "sim/não" como resposta, como por exemplo: "O Linux é um sistema operacional multitarefa?". Para esses tipos de pergunta não há informações ou fatos presentes na resposta (exemplo: data, conceito, pessoas, etc.), na realidade a partir de uma informação localizada no texto é possível concluir a resposta (WAHLSTER *et al.*, 1983).

A arquitetura é dividida em quatro módulos, a saber: análise da pergunta, seleção e extração da resposta, base de conhecimento e geração da resposta. Cada módulo possui objetivos bem definidos e que detalharemos a seguir:

- Análise da pergunta: Este módulo é responsável por examinar a pergunta em linguagem natural e gerar uma consulta (pergunta analisada) que possibilite a seleção dos documentos candidatos a responder a pergunta. As principais atividades deste módulo são: extrair as palavras chaves, lematizar as palavras, remover as palavras irrelevantes, enriquecer a pergunta com sinônimos da

Wordnet, realizar inferências nos conceitos da ontologia, resolver o tipo semântico da pergunta e obter a pergunta resolvida do banco AIML.

- Seleção e extração das respostas: Este módulo é responsável por selecionar os documentos candidatos da web e extrair as respostas. A seleção do documento e a extração das respostas candidatas são baseadas na pergunta analisada. É realizada uma verificação na resposta candidata a fim de torná-la uma resposta factível de retorno. O agente nesta arquitetura se comporta como um gestor das atividades executadas no módulo de seleção e extração da resposta. Assim que uma pergunta é enviada, o agente percebe que é necessário coordenar as atividades de resolução da resposta.

- Base de conhecimento: Este módulo é responsável por organizar e gerenciar consultas à ontologia, à base AIML e à web. Para trabalhos futuros pretendemos utilizar o componente TEXTCOMON (ZOUAQ *et al.*, 2009) para gerar novas ontologias que agreguem novos conhecimentos e inferência na ontologia da arquitetura. A base AIML foi estendida do trabalho desenvolvido por Teixeira (2005) com o fim de aprimorar o retorno da resposta.

- Geração das respostas: Este módulo gera a resposta em um formato textual adequado ao entendimento do usuário. Para trabalhos futuros pretendemos tornar esse módulo interoperável com outros sistemas, ou seja, permitir a troca de perguntas e respostas com outros sistemas por meio de serviços semânticos.

3.2 AS ATIVIDADES

Nesta seção descreveremos todas as atividades e passos envolvidos no funcionamento da arquitetura proposta neste trabalho. A seguir os passos são:

1. O sistema recebe a pergunta em linguagem natural e inicia análise a fim de formar uma consulta para o módulo de seleção e extração das respostas;
2. Na análise da pergunta são realizados os seguintes tratamentos:
    - As *stopwords* são eliminadas da pergunta. As *stopwords* são palavras insignificantes para a recuperação de informação.

- Os conceitos da ontologia são identificados na pergunta por meio de um reconhecedor de entidades. Esse reconhecedor é um dicionário preenchido automaticamente com conceitos da ontologia.
- As palavras simples (que não são conceitos da ontologia) são lematizadas.

3. Por meio de um agente, a pergunta analisada é consultada no banco AIML.
4. Se a resposta for encontrada no banco AIML, então a resposta é retornada ao usuário e o sistema finaliza. Se não for encontrada é iniciado o processo de resolução da pergunta. A base AIML se prefigura como uma memória, ou seja, armazena as perguntas que foram resolvidas para utilizá-las novamente quando for necessário.
5. As palavras simples (que não são conceitos da ontologia) são etiquetadas, cada qual com sua classe gramatical correspondente. A etiqueta será útil para seleção dos substantivos nos passos consecutivos do sistema.
6. Os conceitos são enviados para a ontologia, a fim de raciocinar e retornar outros conceitos subentendidos. Por exemplo: Se o conceito “multitarefa” for enviado para a ontologia um dos conceitos produzidos pela inferência lógica (raciocínio) é: “Linux”. Isso ocorre porque existem as seguintes sentenças lógicas que permitem a inferência: (1) Linux é um sistema operacional, (2) Linux compartilha seus recursos com vários aplicativos, (3) Sistemas Operacionais que compartilham recursos são multitarefa. Esses conceitos subentendidos permitem a expansão da consulta e uma melhora na extração da resposta. A consulta é formada por: (1) conceitos alvos do domínio, (2) palavras significativas, mas que não são conceitos, (3) conceitos subtendidos, (4) sinônimos.
7. Outra forma de expandir a consulta com o fim de melhorar a recuperação é selecionar os sinônimos. São selecionados os substantivos, pois são as palavras com maior peso caracterizador sobre as coisas do mundo real. Nessa fase é estabelecida interpretação singular do sentido da palavra, ou seja, buscamos os sinônimos coerentes com o contexto da palavra. Por exemplo: o substantivo “função” possui inúmeros significados na *Wordnet*.
8. O tipo semântico da pergunta é resolvido por meio de padrões encontrados na pergunta. O tipo semântico será útil para formatar a resposta. Por exemplo: Para a pergunta: Quem inventou o Linux? O tipo semântico é uma Pessoa. Então a resposta deve ter um formato compatível com o tipo semântico.

9. A pergunta analisada ou consulta expandida é enviada para a Web com o fim de selecionar as páginas candidatas.
10. As páginas candidatas são transformadas em documentos passíveis de extração, ou seja, toda formatação inútil é extraída, tais como: estilo CSS, HTML.
11. Os trechos da resposta são extraídos e classificados com uma pontuação (grau de relevância). Nesse momento a resposta não é a final, e sim uma resposta candidata (provável). Essa atividade é auxiliada por um algoritmo.
12. As respostas candidatas passam por um processo de confirmação para verificar se são respostas factíveis. Esse processo de confirmação é realizado por meio de uma técnica de implicação textual (RTE - *Recognizing Textual Entailment*). O RTE é definido como uma tarefa de determinação se um dado pedaço de texto T implica em outro pedaço de texto H, chamado Hipótese. O RTE avalia cada resposta candidata em relação à pergunta.
13. As respostas são enviadas para o módulo de geração da resposta que as formata de acordo com o tipo semântico.
14. A resposta é retornada para o usuário. Para trabalhos futuros o usuário terá a opção de emitir opinião sobre a qualidade da resposta.
15. A resposta é enviada para um especialista do domínio para avaliação. De acordo com a pontuação da avaliação o sistema grava a resposta na base AIML por meio do editor. O editor é um componente que gerencia os acessos externos a base AIML. A atividade de avaliação do especialista é um trabalho futuro pretendido por este trabalho.

3.3 COMPARAÇÃO COM O ESTADO DA ARTE

Nesta seção mostraremos as diferenças conceituais entre o sistema proposto nesta dissertação e os sistemas referenciados no estado da arte. A seguir elencamos os seguintes pontos:

- De forma geral as ontologias nos sistemas pergunta-resposta são utilizadas como fontes das respostas (DAMIJANOVIC *et al.*, 2010), mas para perguntas que não recebem respostas singulares, exemplo: O que são sistemas multitarefa? Utilizando apenas as ontologias é difícil responder perguntas que exigem definições ou explicações. A ontologia na arquitetura proposta nesta dissertação

funciona como uma parte do cérebro do sistema. A partir de um conceito presente na pergunta é possível aplicar inferências lógicas para extração de novos conceitos, e dessa forma, enriquecer a consulta a fim de melhorar a extração da resposta.

- Normalmente sistemas de pergunta-resposta iniciam todo processo de resolução da pergunta ao invés de verificar se a pergunta já foi resolvida anteriormente, buscando em alguma base de conhecimento. A classe de sistema que costuma utilizar essas práticas são os *helpdesk* (WANG *et al.*, 2011), mas poucos utilizam bases próprias para *chatterbots*. Na arquitetura proposta nesta dissertação é utilizada uma base AIML, própria para *chatterbot*, que é construída à medida que as perguntas são resolvidas. Dessa forma a base pode ser aproveitada para um *chatterbot* interagir com o usuário, fornecendo a resposta. Além disso, antes da resposta ser incluída na base AIML especialistas do domínio confirmam a veracidade da resposta, dessa forma a base AIML é expandida.
- Muitos sistemas na literatura retornam a resposta sem antes verificar e testar, ou seja, verificar se a resposta é realmente válida para perguntas. Alguns poucos sistemas até realizam essa verificação, mas poucos utilizam a técnica RTE. Na arquitetura proposta nesta dissertação utilizamos a técnica RTE para validação da resposta. Essa técnica é bastante consolidada e apoia-se por mostrar bons resultados (IFTENE, 2009) (FERRÁNDEZ *et al.*, 2011).
- A arquitetura proposta também utiliza um recurso denominado *Word Sense Disambiguation* (NAVIGLI *et al.*, 2011) para buscar os sinônimos corretos para o contexto da pergunta. Dentro do domínio de WSD utilizamos a técnica de aprendizagem de máquina supervisionada.

3.4 CONCLUSÃO

Este capítulo concentra-se no funcionamento conceitual da arquitetura do sistema de pergunta-resposta proposto. Por enquanto tudo considerado existente está no domínio das ideias, apoiado por pesquisas na literatura. O objetivo da arquitetura é responder perguntas *WH-question* (O que, Quem, Quando, Onde, Quais, Quem) da língua inglesa para o domínio de sistemas operacionais. A arquitetura está dividida em quatro partes bem definidas: análise da pergunta, seleção e extração da resposta, base de

conhecimento e gerador da resposta. Para que o sistema alcance o objetivo, retornar a resposta, propomos a utilização de bases de conhecimento: ontologias e AIML.

CAPÍTULO 4 PROPOSTA DA SOLUÇÃO TECNOLÓGICA

Neste capítulo descrevemos todos os componentes tecnológicos necessários para que a arquitetura conceitual possa realizar as atividades necessárias para resolução da pergunta.

Na Seção 4.1 Visão Geral, ilustramos a arquitetura conceitual juntamente com seus componentes tecnológicos e descrevemos, de forma sucinta, a ação dos componentes no fluxo do sistema.

Na próxima seção apresentamos os componentes tecnológicos utilizados na arquitetura deste trabalho. Nas subseções: 4.2.1 O Protégé, 4.2.2 *Crawler*, 4.2.3 RTE , 4.2.4 LingPipe, 4.2.5, Lucene e 4.2.6 AIML, Detalhamos as características e funções de cada componente tecnológico

4.1 VISÃO GERAL

Em tese, a ciência básica tem como objetivo o puro conhecimento de um determinado assunto seja ele qual for. A ciência aplicada surge quando aparece a oportunidade de, com os conhecimentos científicos adquiridos, resolver um problema prático sem cogitar das implicações socioeconômicas de sua solução. Quando tais implicações são levadas em conta é que surge a tecnologia, como utilização, e não simples aplicação, de conhecimentos científicos do problema técnico (VARGAS, 2011).

Neste capítulo descrevemos todos os componentes tecnológicos necessários para que a arquitetura conceitual possa realizar as atividades necessárias para resolução da pergunta. A seguir ilustramos a arquitetura conceitual juntamente com os componentes tecnológicos:

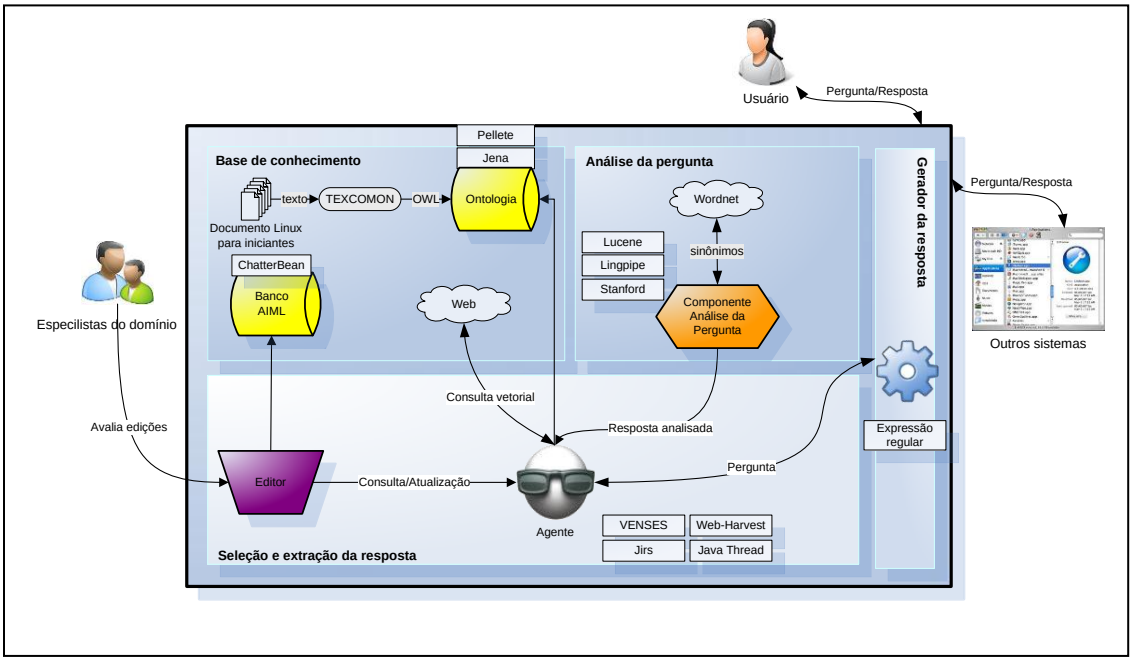

Figura 4.1 – Arquitetura tecnológica do sistema de pergunta-resposta com os componentes.

O sistema foi desenvolvido na linguagem Java e apoiado por vários outros componentes, tais como: *chatterbean*, *lingpipe*, entre outros. Antes de iniciar o desenvolvimento, a ontologia deve ser construída ou partir de algum componente gerador tal como o TEXTCOMON (ZOUAQ *et al.*, 2009), ou manualmente pela utilização de alguma ferramenta tal como: *Protégé*. Para trabalho futuros propomos o uso do TEXTCOMON tanto para gerar a ontologia inicial, como agregar novos conceitos gerados. Neste trabalho desenvolvemos uma ontologia particular utilizando o *Protégé*.

Nesta seção descreveremos todas as atividades e passos envolvidos no funcionamento da arquitetura proposta com os componentes tecnológicos. A seguir, os passos são mostrados na Tabela 4.1:

Tabela 4.1 – Atividades da arquitetura com os componentes tecnológicos.

Módulo	**Descrição**	**Componentes Tecnológicos**
Geração da resposta	O sistema recebe a pergunta em linguagem natural	Linguagem Java
Análise da pergunta	O sistema inicia análise na pergunta a fim de formar uma consulta para o módulo de seleção e extração das respostas.	Linguagem Java

Módulo	**Descrição**	**Componentes Tecnológicos**
Análise da pergunta	As palavras irrelevantes (*stopwords*) são eliminadas da pergunta.	Lucene
Análise da pergunta	Os conceitos da ontologia são identificados na pergunta. Esse recurso é conhecido como reconhecimento de entidades (NER). (Ver na Seção 4.2.4)	Lingpipe
Análise da pergunta	As palavras (exceto os conceitos da ontologia) são lematizadas.	Stanford CoreNLP
Análise da pergunta	A pergunta analisada é enviada para o banco AIML através de um agente.	Chatterbean
Base de conhecimento	Se a resposta for encontrada, então a resposta é retornada e o sistema finaliza. Se não for encontrada o agente inicia o processo de resolução da pergunta. A base AIML se prefigura como uma memória, ou seja, armazena as perguntas que já foram resolvidas para utilizá-las novamente quando for necessário.	Chatterbean
Análise da pergunta	As palavras simples (que não são conceitos da ontologia) são etiquetadas, cada qual com sua classe gramatical correspondente. Esse recurso é conhecido como etiquetagem do discurso. (Ver na Seção 4.2.4)	Lingpipe
Análise da pergunta	Os conceitos são enviados para a ontologia, a fim de raciocinar e retornar outros conceitos subentendidos.	Pellete, Jena
Análise da pergunta	São consultados na *Wordnet* os sinônimos dos substantivos lematizados. A extração dos sinônimos é realizada de acordo com o contexto da pergunta. Por exemplo: A palavra "função" possui sete entradas no dicionário *Wordnet*, mas de acordo com a pergunta "Quais são as funções básicas de um sistema operacional?", os sinônimos	Java, Lingpipe

Módulo	**Descrição**	**Componentes Tecnológicos**
Análise da pergunta	selecionados devem ser: “propósitos” e “objetivos”. Para realizar essa atividade utilizamos a biblioteca Lingpipe, que emprega aprendizagem de máquina supervisionada. Esse recurso é conhecido como Desambiguação dos sentidos das palavras. (Ver na Seção 4.2.4)	Java, Lingpipe
Análise da pergunta	O tipo semântico da pergunta é retornado	Expressão regular
Base de conhecimento	A pergunta analisada é enviada para a Web com o fim de selecionar as páginas candidatas.	Web-Harvest
Seleção e extração da resposta	As páginas candidatas são transformadas em documentos passíveis de extração, ou seja, toda formatação é removida.	HTMLCleaner[10], Jtidy[11]
Seleção e extração da resposta	São extraídos trechos ou passagens das páginas candidatas. Essas passagens são consideradas como prováveis resposta à pergunta. As passagens retornadas são listadas com uma probabilidade associada que indica o grau de relevância com a pergunta analisada.	Jirs
Seleção e extração da resposta	As respostas candidatas passam por um processo de confirmação para verificar se são respostas factíveis à pergunta. Esse processo de confirmação é realizado por meio de uma técnica de implicação textual (RTE - *Recognizing Textual Entailment*). O RTE é definido como uma tarefa de determinar se um dado pedaço de texto T implica em outro pedaço de texto H, chamado Hipótese. O RTE avaliaria cada resposta candidata em relação à pergunta	Venses

[10] http://htmlcleaner.sourceforge.net/

[11] http://jtidy.sourceforge.net/

Módulo	Descrição	Componentes Tecnológicos
Geração da resposta	A resposta candidata de maior probabilidade é enviada para o módulo de geração da resposta que a formata de acordo com o tipo semântico.	Expressão regular
Geração da resposta	A resposta é retornada para o usuário.	Linguagem Java
Geração da resposta	A resposta é enviada para um especialista do domínio para avaliação. De acordo com a pontuação da avaliação o sistema grava a resposta na base AIML.	Linguagem Java, ChatterBean

4.2 OS COMPONENTES TECNOLÓGICOS

A fim de aprofundar as funcionalidades dos principais componentes da arquitetura detalharemos nas seções a seguir.

4.2.1 Ontologias

Ontologias são usadas para capturar conhecimento sobre algum domínio de interesse. Uma ontologia descreve conceitos no domínio e também as relações mantidas entre esses conceitos. Diferentes linguagens de ontologias fornecem diferentes facilidades. A mais recente linguagem de desenvolvimento de ontologia é OWL da W3C. A OWL tem um rico conjunto de operadores – exemplo: interseção, união e negação. Baseia-se em um modelo lógico que torna possíveis conceitos serem definidos. Conceitos complexos podem ser construídos a partir de definições de conceitos mais simples. Além disso, o modelo lógico permite utilizar um raciocinador que pode verificar se há ou não sentenças e definições na ontologia que são mutuamente consistentes e pode reconhecer quais conceitos se ajustam em quais definições. O raciocinador ajuda a manter uma hierarquia correta. Isso é útil particularmente quando lidamos com casos onde classes podem ter mais de um pai (HORRIDGE *et al.*, 2011; GÓMEZ-PÉREZ *et al.*, 2007).

Uma ontologia define um vocabulário comum para pesquisadores que precisam compartilhar informações de um domínio. Porque alguém desenvolveria uma ontologia? (NOY *et al.*, 2001)

- Para compartilhar entendimento comum da estrutura da informação entre pessoas ou entre agentes;
- Para permitir reutilizar conhecimento do domínio;
- Para fazer hipóteses explícitas do domínio;
- Para separar conhecimento do domínio do conhecimento operacional;
- Para analisar conhecimento do domínio.

Para o desenvolvimento da ontologia desta dissertação utilizamos as seguintes diretrizes, exibidas na Tabela 4.2.

Tabela 4.2 – Componentes utilizados no desenvolvimento da ontologia.

Ferramenta	Protégé
Raciocinador	Pellete
Biblioteca Java	Jena
Domínio modelado	Sistemas operacionais
Linguagem da ontologia	OWL 2.0

A extensão do escopo do domínio está ligada ao fato que uma ontologia descreve conhecimento para uma comunidade de usuários em virtude de significados acordados de um vocabulário usado (GUARINO, 1998). Portanto a modelagem da ontologia desenvolvida neste trabalho está baseada nas conceituações de: (MACHADO *et al.*, 2008) (TANENBAUM *et al.*, 2006). A Figura 4.2 ilustra a tela inicial de configuração do Protégé da ontologia de sistema operacionais.

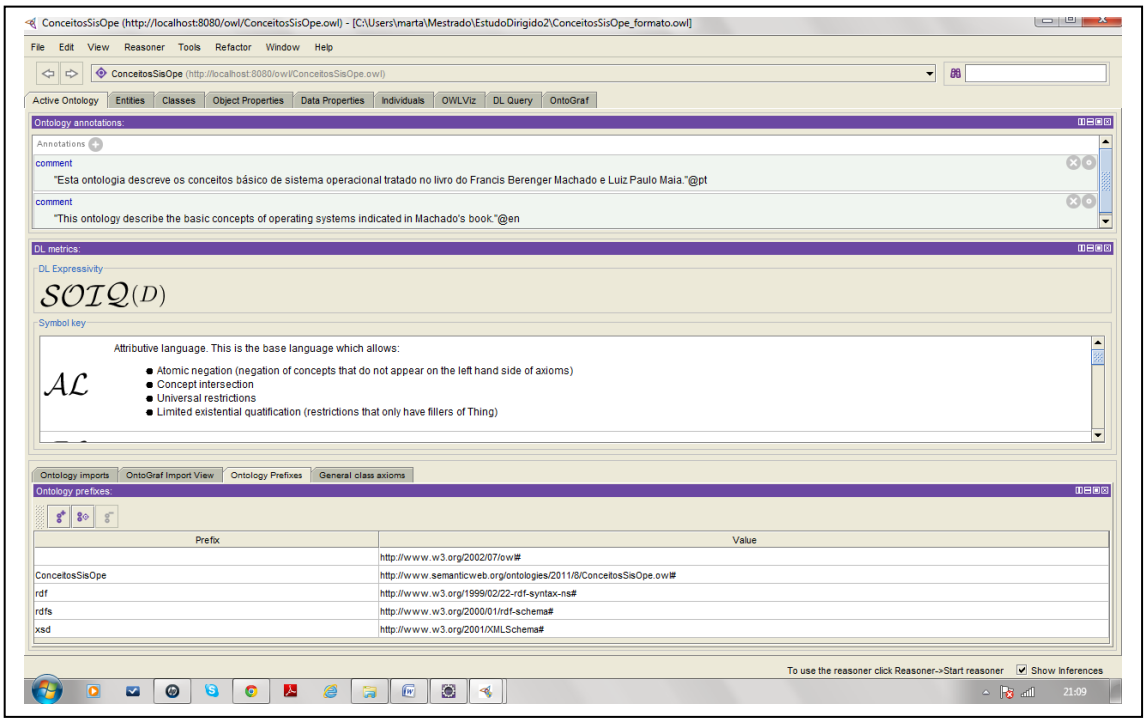

Figura 4.2 – Tela inicial do Protégé com as configurações iniciais da ontologia de sistemas operacionais.

A identificação das questões de competência faz parte da boa práxis da engenharia de ontologia (FALBO *et al.*, 2002). As questões de competência podem ser consideradas requisitos que são feitas na forma de perguntas e que uma ontologia deve responder (GRUNINGER *et al.*, 1995). Todas as questões de competências estão listadas no Apêndice A. Na Figura 4.3 ilustramos a ontologia desenvolvida no *Protégé* usando a linguagem OWL.

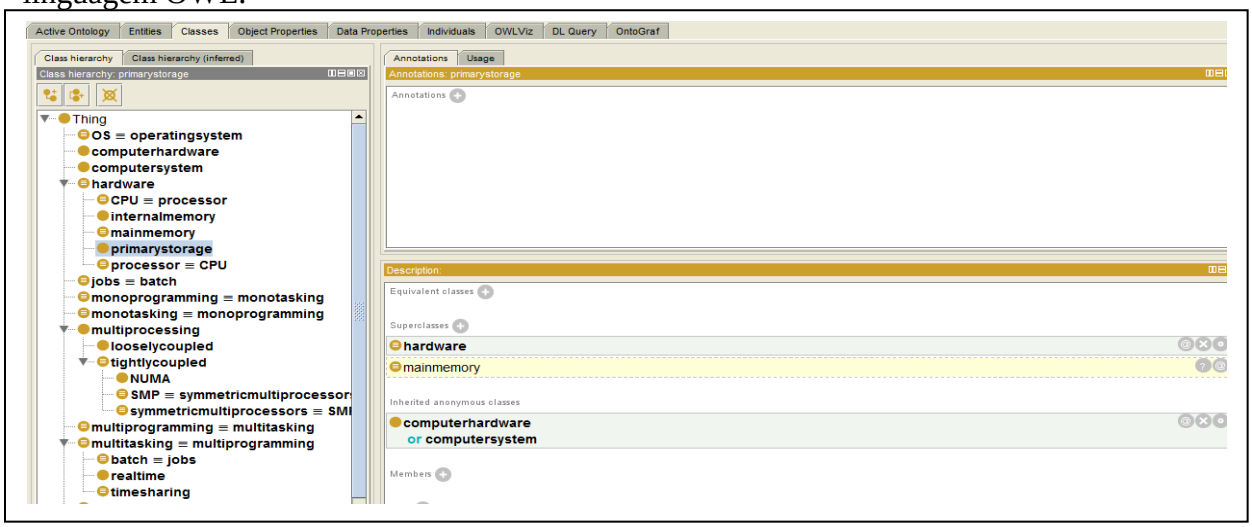

Figura 4.3 – Tela do Protégé com a ontologia de sistemas operacionais.

A linguagem OWL é baseada em um modelo lógico (HORRIDGE *et al.*, 2011; GÓMEZ-PÉREZ *et al.*, 2007) que descreve o conhecimento do domínio utilizando principalmente: classes, relacionamentos (com propriedades matemáticas. Exemplo: simétrica, transitivas, etc.) e restrições (existências e universais). A partir do modelo é possível realizar inferências lógicas e extrair novas informações.

Observe a Figura 4.4, o conceito "*Cache*" foi inferido automaticamente como classe disjunta do conceito "*Nonvolatilememory*". Essa inferência lógica é realizada pelo componente Pellet.

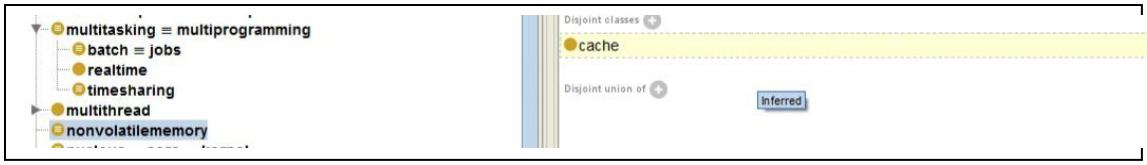

Figura 4.4 – Tela do Protégé com a ontologia de sistemas operacionais.

Para explicar um pouco mais como as inferências lógicas são realizadas, ilustramos na Figura 4.5 os principais componentes da arquitetura do Pellet. O Pellet, em sua essência, é um raciocinador de lógica descritiva baseado no algoritmo Tableaux (SIRIN *et al.*, 2007).

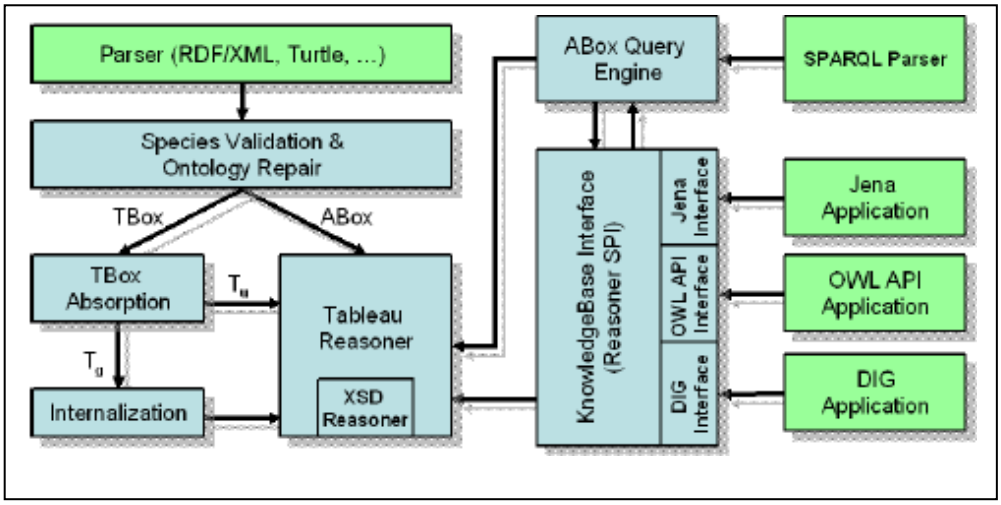

Figura 4.5 – Os principais componentes do Pellet (Fonte: SIRIN et al., 2007)

4.2.2 Crawler

A necessidade de um *crawler* em um ambiente baseado na web é para armazenar e colecionar informações distribuídas. Na intranet de grandes corporações, é difícil visitar centenas de sites e manualmente coletar informações de todas as páginas. Ao invés disso é utilizado um *crawler* para visitar todas as páginas que podem ser acessadas a partir de um conjunto de URL. Os termos *bot*, *spider* e *robot* são frequentemente usados para descrever um *crawler* (KONCHADY, 2008).

A ideia básica do funcionamento do *crawler* é bastante simples. Cada página web tem uma URL (*Universal Resource Locator*) que identifica a localização da página na web. Um típico *crawler* leva uma ou mais URLs como entrada para formar uma lista inicial de URLs. O *crawler* então repete os dois seguintes passos até que nenhuma URL nova possa ser encontrada ou que todas as páginas tenham sido buscadas: (1) Pega a próxima URL da lista de URLs, estabelecendo conexão com o servidor onde a página reside, e busca a página no servidor por meio de um pedido HTTP. (2) Extrai novas URLs de cada busca à página web e as adiciona na lista (MENG *et al.*, 2010).

Neste trabalho o principal objetivo é utilizar um *crawler* para realizar buscas em páginas específicas do domínio das perguntas submetidas. Ao invés de buscar em toda a web, pode-se customizar um *crawler* para buscar e extrair páginas específicas ao conteúdo perguntado. Utilizamos neste trabalho às bibliotecas Java do Web-Harvest[12] para realizar as atividades de um *crawler*.

4.2.3 RTE

Um fenômeno básico da linguagem natural é a variabilidade de expressões semânticas, onde o mesmo significado pode ser expresso por, ou implicado de textos diferentes. Esse fenômeno pode ser considerado como um problema duplo da ambiguidade da linguagem, formando juntamente um mapeamento de muitos-para-muitos entre as expressões da linguagem e os significados. Muitas aplicações que utilizam processamento de linguagem natural, tal como sistemas pergunta-resposta, extração de

[12] http://web-harvest.sourceforge.net/index.php

informação, sumarização de documentos, avaliação de tradução automática, podem utilizar os recursos de implicação textual (DAGAN *et al.*, 2009).

Dentro do framework RTE (*Recognising Textual Entailment* – Identificação de Implicação Textual), um texto T é dito como implicação de em outro texto hipótese H se a verdade de H pode ser implicada de T. Isso significa que a maioria das pessoas concordaria que o significado de T implica H, Um pouco mais formal, dizemos que T implica H quando alguma representação de H pode ser "combinada" com alguma (ou parte de uma) representação de T, em algum nível de granularidade e abstração (IFTENE, 2009).

Por exemplo, um sistema de pergunta-resposta identifica textos que combinem com respostas hipotéticas. Dada uma pergunta "Quem pintou O Grito?", O texto "A mais famosa pintura da Noruega, O Grito de Edvard Munch" implica a hipotética resposta "Edvard Munch pintou O Grito". Similarmente para consultas de recuperação da informação que combinam conceitos semânticos e relações que devem ser implicados de documentos relevantes recuperados (DAGAN *et al.*, 2009).

No passado as competições de implicação textual (RTE1 em 2005, RTE3 em 2006 e RTE3 em 2007) foram organizadas por PASCAL (*Pattern Analysis Statistical Modelling and Computational Learning*). Em 2008, a quarta edição, o desafio foi organizado dentro da conferência de análise de textos. A conferência de análise de textos (TAC) é uma série de *workshops* de avaliação, organizado para encorajar a pesquisa no processamento de linguagem natural e aplicações relacionadas (IFTENE, 2009).

Os sistemas RTE demonstraram avanço com o tempo, com níveis de precisão alcançando de 50% a 60% no ano de 2005 (17 submissões - RT1), de 53% a 75% em 2006 (23 submissões - RTE2), de 49% a 80% em 2007 (26 submissões – RTE3) e de 45% a 74% no RT4 (DAGAN *et al.*, 2009).

A fim de dirimir qualquer dúvida, é crucial explicar a diferença entre inferência e implicação, pois são coisas diferentes. A inferência e implicação são processos psicológicos (mental ou intelectual) que conduzem a possíveis mudanças na crença ou possíveis mudanças nos planos e intenções. A implicação é mais uma relação direta entre proposições. Determinada proposição implica outra proposição quando e apenas quando, se as primeiras proposições são verdadeiras, então a última também é verdadeira (GABBAY *et al.*, 2002).

Podemos dizer que "A, B, e C implica D". Outra coisa bastante diferente é dizer, "Se você acreditar em (julga verdadeiro) A, B, e C, você deve ou pode inferir D". A primeira não diz nada especial sobre crença ou qualquer outro estado psicológico, e também não diz qualquer restrição sobre o que alguém "deve" ou "pode" fazer. O primeiro exemplo pode ser verdadeiro sem o segundo ser verdadeiro. Então percebemos que a implicação impõe limites. Uma pessoa poderia acreditar em A, B, e C sem ter qualquer razão para crê em D. Além disso, uma pessoa que acredita em A, B, e C e percebe que A, B e C implica em D pode também ter boas razões para acreditar que D é falso. E mesmo alguém acreditando em A, B, e C, que perceba que A,B, e C implique em D, e que alguém tem nenhuma razão para pensar que D é falso, pode não ter nenhuma razão para inferir D. Essa pessoa poderia ser desinteressada se D é verdadeiro ou falso. Muitas coisas triviais seguem uma crença sem ter qualquer motivo para inferi-los (GABBAY *et al.*, 2002).

A implicação realizada pelo componente tecnológico RTE desta dissertação é puramente sintática, ou seja, se baseia nas relações gramaticais para concluir se um pedaço de texto implica em outro texto, ao passo que inferência realizada pela ontologia é lógica ou matemática. Entretanto existem outros componentes tecnológicos RTE que realizam a implicação baseado em outras técnicas, como por exemplo: lógica de primeira ordem (Nutcracker[13]). O componente tecnológico utilizado para realizar a implicação textual é o VENSES. A proposta de utilização do RTE neste projeto é verificar se a resposta provável implica na pergunta analisada. Se o retorno for verdadeiro, a resposta será retornada.

[13] http://svn.ask.it.usyd.edu.au/trac/candc/wiki/nutcracker

4.2.4 LingPipe

O *Lingpipe* é uma biblioteca para processamento de textos usando linguística computacional. No projeto desta dissertação utilizamos os seguintes recursos do lingpipe: reconhecimento de entidades (NER), desambiguação dos sentidos das palavras (WSD) e etiquetagem.

- Reconhecimento de entidades (NER) – É um processo de encontrar menções de coisas especificadas no texto. Por exemplo: Um reconhecedor de entidades para o inglês pode encontrar uma entidade pessoa "John J. Smith" e a localidade "Seattle" no texto "John J. Smith mora em Seattle.". A *Lingpipe* possui disponível um vasto corpus biomédico para encontrar entidades. Por exemplo: "Fibrinogênio humano" e "Trobina" são menções de genes na frase: "Fibrinogênio humano nativo foi levado à coagulação por adição de trombina." O reconhecimento de entidades no Lingpipe envolve treino supervisionado de modelos estatísticos ou métodos mais diretos como busca em dicionários ou expressões regulares.
- Etiquetagem do discurso – A etiquetagem do discurso é um processo por meio do qual *tokens* são sequencialmente etiquetados com informações sintáticas, tais como "verbo" ou "gerúndio" ou "conjunção subordinada".
- Desambiguação dos sentidos das palavras – É o processo de determinar qual o sentido de uma palavra polissêmica é pretendido para um dado contexto. Exemplo: a palavra "função" possui sete significados no dicionário, qual desses representa o contexto determinado pela frase seguinte: "Quais as funções básicas do sistema operacional?". A desambiguação do sentido é realizada por meio de aprendizagem de máquina supervisionada que pode utilizar os seguintes tipos de classificadores: modelos de linguagem n-gram, Naive Bayes, entre outras (SOMAN *et al.*, 2009). Este trabalho utiliza o classificador modelo de linguagem n-gram.

4.2.5 Lucene

Lucene é uma biblioteca para recuperação da informação. Recuperação da informação refere-se ao processo de busca por documentos, por informação dentro dos documentos ou metadados sobre documentos. Lucene permite adicionar novas possibilidades na busca, por exemplo: filtros, análise de palavras, entre outros. É uma biblioteca livre e implementada em Java (HATCHER *et al.*, 2009). A biblioteca Lucene foi usada no projeto desta dissertação com o fim de remover as palavras não significativas (*stopwords*).

As *Stopwords* são palavras funcionais, como artigos, conectivos e preposições, que não carregam significado e, portanto podem ser ignoradas (BAEZA-YATES *et al.*, 2000; GONZALEZ *et al.*, 2003). Com tal eliminação, corre-se, entretanto, o risco de perder a estrutura composicional de expressões. As preposições, por exemplo, podem exercer papel composicional significativo. Entretanto, como termos isolados perdem significado ao contrário de outras categorias gramaticais como o substantivo (GONZALEZ *et al.*, 2003).

4.2.6 AIML

Chatterbots podem facilitar o processo de interação humano-computador e também permitem explorar e influenciar o comportamento do usuário. Recentes estudos têm mostrado a importância dessa figura para melhorar o desempenho dos sistemas de computador (GALVÃO *et al.*, 2004). Por isso, é de grande interesse no projeto desta dissertação o uso de *bot* e base AIML para auxiliar na busca da resposta.

O AIML ou, *Artificial Intelligence Markup Language* é uma linguagem que permite descrever o conhecimento que será manipulado por *chaterbots* baseado na tecnologia A.L.I.C.E. O AIML foi desenvolvido pela comunidade de software *Alicebot* durante 1995-2000 e foi originalmente adaptada da gramática do XML.

O AIML descreve uma classe de objetos que são compostos de unidades chamadas tópicos e categorias, na qual contém dados analisados e não analisados. Os dados analisados são compostos de dados de caracteres ou elementos AIML. Os elementos AIML encapsulam o conhecimento estímulo-resposta (ABRAHAM *et al.*, 2011). Na

Figura 4.5, a seguir temos um exemplo do trecho AIML armazenado na base do projeto desta dissertação.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<aiml>
<category>
<pattern>BASIC FUNCTION OPERATING SYSTEM</pattern>
 <template>The function basic are: Operating system controls and coordinates the use of the hardware among the various applications programs for various uses. Operating system acts as resource allocator and manager. Since there are many possibly conflicting requests for resources the operating system must decide which requests are allocated resources to operating the computer system efficiently and fairly. Also operating system is control program which controls the user programs to prevent errors and improper use of the computer. It is especially concerned with the operation and control of I/O devices.
</template>
</category>
</aiml>
```

Figura 4.6 – Código AIML usado no projeto.

4.2.7 JIRS

JIRS é uma biblioteca Java específica para recuperação de passagens. O JIRS é capaz de achar passagens usando os n-gramas da pergunta e calcular a similaridade da passagem extraída. O JIRS usa um tradicional sistema para recuperação de passagens, como um primeiro passo. Então busca todos os possíveis n-gramas da pergunta na recuperação das passagens e, por final classifica com um peso dos n-gramas que foram encontrados nas passagens. Na Figura 4.7 podemos observar a principal estrutura do JIRS.

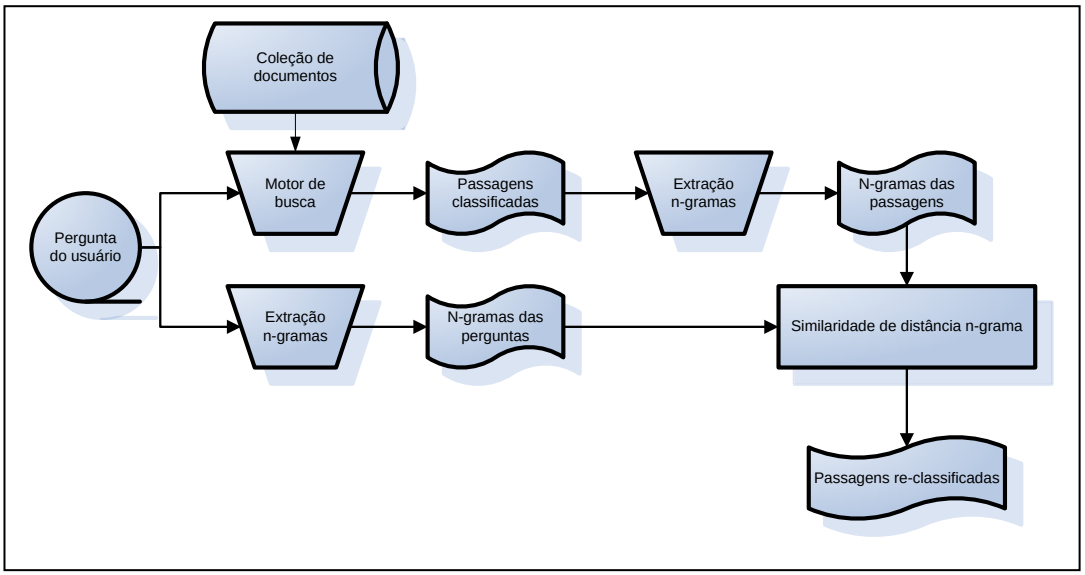

Figura 4.7 – Estrutura principal do JIRS.

4.2.7.1 Explicando o modelo n-grama

Regida pela teoria da probabilidade, o estudo das variáveis estocásticas introduz diversos tipos de medidas, como distribuição, frequência, esperança e variância, e também modelos. Um modelo probabilístico que analisa sequência de palavras (modelo n-grama ou modelo de Markov de ordem n-1) supondo que as n-1 palavras antecedentes afetam a probabilidade das próximas palavras. No âmbito da teoria da informação, a estatística provê mecanismos para indicar quanta informação ou quanta incerteza temos em relação a um evento, ou, ainda, qual o grau de associação de eventos co-ocorrentes. O modelo de n-grama é um método estatístico que utiliza mapeamento de co-ocorrência, basicamente considerando a frequência de ocorrência dos termos em coleções de documentos. O n-grama baseia-se em similaridade entre palavras encontradas em janelas de texto (GONZALEZ *et al.*, 2003).

O módulo de “Motor de busca” encontra as passagens (exemplo: pedaços de textos) com as palavras chaves da pergunta. Cada passagem retornada pelo “Motor de busca” é classificada com um peso. Essa passagem ponderada é igual à soma de todos os termos (palavras) ponderados da pergunta. O termo ponderado é calculado por:

$$w_k = 1 - \frac{\log(n_k)}{1 + \log(N)}$$

Figura 4.8 – Fórmula para ponderação dos termos.

Onde n_k é o número de passagens na qual os termos t_k ocorrem e N é o número total de passagens. De acordo com a equação da Figura 4.8, cada termo tem peso diferente dependendo de sua relevância. Portanto, as passagens retornadas no topo da classificação são mais relevantes. Com as *m* passagens mais relevantes, o sistema extrai o(s) 1-grams, 2-grams e assim por diante (onde n é o número de termos na pergunta). Em paralelo, os n-gramas da pergunta é extraído. Assim, a pergunta e o conjunto de passagens dos n-gramas são comparados usando modelo de distância n-grama. Esse modelo encontra a estrutura da pergunta na passagem e determina um valor alto de similaridade para aquelas passagens que contém mais estruturas agrupadas. Esse valor de similaridade é calculado por:

$$Sim(p,q) = \frac{1}{\sum_{i=1}^{n} w_i} \cdot \sum_{\forall x \in P} h(x) \frac{1}{d(x, x_{max})}$$

Figura 4.9 – Fórmula para calcula da similaridade.

Seja Q o conjunto de n-grama de p composto apenas por termos da pergunta. Portanto, definimos P = {x_1, x_2, ..., x_M} assim como o subconjunto ordenado de Q que preenche as seguintes condições:

$$\forall x_i \in P :$$
$$h(x_i) \geq h(x_{i+1}) \quad i \in \{1, 2, ..., M-1\}$$
$$\forall x, y \in P : x \neq y \Rightarrow T(x) \bigcap T(y) = \emptyset$$
$$\min_{x \in P} h(x) \geq \max_{y \in Q \setminus P} h(y)$$

Figura 4.10 – Regras e condições.

Onde R(x) é o subconjunto dos termos de n-grama de x e h(x) é a função definida por:

$$h(x) = \sum_{k=1}^{j} w_k$$

Figura 4.11 – Fórmula de h(x).

Onde w_1, w_2, ..., $w_{|w|}$ são termos ponderados de n-grama de x e são calculados pela equação da figura 4.11. Esses pesos determinam o estímulo para aqueles termos que não aparecem com muita frequência em uma determinada coleção. Além disso, o peso deve também discriminar os termos que ocorrem frequentemente na coleção (ex: *stopwords*).

O d(x, x_{max}) é um fator de distância entre o n-grama x e o n-grama x_{max} e é calculado por:

$$d(x, x_{max}) = 1 + k \cdot \ln(1 + L)$$

Figura 4.12 – Fator de distância

Onde L é o número de termos entre o n-grama x_{max} (x_{max} é o n-grama com o maior peso calculado em (3)) e o n-grama x da passagem. Se existe mais de um n-grama x na passagem, é escolhido o mais próximo. A fim de medir o grau de importância do fator de distância na equação de similaridade, foi introduzido a constante k. Em experimentos anteriores foi determinador que o melhor valor é 0.1 As outras constantes são usadas para evitar valores infinitos quando L é igual à zero.

A biblioteca JIRS é totalmente configurável e permite procurar as passagens pelos seguintes modelos: modelo de espaço vetorial e modelos baseado em n-grama (modelos simples (SNM), modelo de termos ponderado e modelo de distância (DNM)). Neste projeto de dissertação estamos usando o modelo de distância n-grama (DNM). Toda pesquisa relatada nesta seção sobre o JIRS foi extraída do artigo de Buscaldi *et al.* (2010).

4.2.8 STANFORD NLP

Stanford CoreNLP fornece um conjunto de ferramentas para análise da linguagem natural que recebe texto da língua inglesa como entrada e determina formas básicas de suas palavras, partes do discurso, sejam nomes de empresa, pessoas, normalização de

datas, horas e quantidades numéricas, e marcam estruturas de sentenças em termos de frases e palavras, e indicam quais sintagmas nominais referem à mesma entidade.

O recurso utilizado da biblioteca Stanford NLP neste projeto de dissertação é a lematização. A lematização é um processo de normalização morfológica que reduz os itens lexicais por meio da conflação[14], a uma forma que procura representar classes de conceitos. Outro processo muito comum de normalização morfológica é o *stemming* (GONZALEZ *et al.*, 2003). As diferenças entre dois processos são:

- *Stemming* – reduz todas as palavras com o mesmo radical a uma forma denominada *stem* (similar ao próprio radical), sendo eliminados afixos oriundos de derivação ou de flexão (em alguns casos, apenas o sufixo de palavras são retirados). Exemplo: "construções" e "construiremos" seriam reduzidas ao *stem* "constru".
- Redução à forma canônica (tratada por alguns autores como *lemmatization*), que, geralmente, reduz os verbos ao infinitivo e os adjetivos à forma masculina singular. Exemplo: "construções" e "construiremos" seriam reduzidos a "construção" e "construir".

4.2.9 Expressões Regulares

Alguns sistemas de recuperação de informação permitem busca por expressões regulares. A expressão regular é um padrão bastante geral construído por caracteres que permitem a recuperação de pedaços de textos correspondentes ao padrão. (BAEZA-YATES *et al.*, 2000). Neste projeto utilizamos as bibliotecas nativas do Java para construir expressões regulares. A Tabela 4.3 exemplifica algumas expressões regulares utilizadas no projeto:

[14] Algoritmos de conflação (*conflation*) são aqueles que combinam a representação de dois ou mais termos num único termo, ou seja, reduzem variantes de uma palavra numa forma única. (JONES *et al.*, 1997)

Tabela 4.3 – Exemplo de expressões regulares

Expressão regular	**Exemplo de pergunta com padrão**	**Tipo semântico**	**Entidade**
^[Hh]ow _DT[a-zA-Z] _CONCEPT	How the main memory of a computer is organized?	Factoid	Reason
^[wW]hat is _CONCEPT	What is CISC processor?	Factoid	Definition

4.3 CONCLUSÃO

Todo conteúdo tratado neste capítulo se resume na escolha das soluções tecnológicas para a arquitetura conceitual (ver Capítulo 3) do sistema de pergunta-resposta proposto. Detalhamos o funcionamento dos principais componentes tecnológicos e de forma geral explicamos o funcionamento da arquitetura.

O módulo “Análise da pergunta” utiliza as seguintes soluções tecnológicas: Ontologias, Lucene, Stanford NLP, Lingpipe e Expressões Regulares. As ontologias serão úteis para expansão da consulta, ou seja, retornar conceitos importantes para a fase da extração da resposta. A biblioteca Lucene elimina as *stopwords*. A biblioteca Stanford NLP lematiza os conceitos. A biblioteca Lingpipe reconhece os conceitos da pergunta e seleciona os sinônimos de acordo com o contexto. E as expressões regulares reconhecem o tipo semântico da pergunta.

O módulo “Seleção e extração da resposta” utiliza as seguintes soluções tecnológicas: RTE, *Crawler*, JIRS e expressão regular. O RTE verifica se o trecho da resposta provável implica na pergunta analisada. O *Crawler* extrai as páginas da web. O JIRS extrai passagens candidatas a resposta. E as expressões regulares formatam as passagens candidatas de acordo com o tipo semântico da pergunta.

O módulo “Base de conhecimento” utiliza as seguintes soluções tecnológicas: Ontologias e AIML. As ontologias estruturam o conhecimento do domínio. E a base AIML armazenam os pares pergunta/resposta.

CAPÍTULO 5 ESTUDO DE CASO

Nos capítulos anteriores apresentamos as bases teóricas e tecnológicas para a construção do trabalho proposto. Seguiremos nesse capítulo com a construção do protótipo, realização do experimento e avaliação da proposta sugerida.

Para avaliar o método proposto foi confeccionado um protótipo que será apresentado na Seção 5.1. Nessa seção será apresentado um exemplo demonstrando o funcionamento do sistema.

Na Seção 5.2 relatamos os experimentos realizados com perguntas extraídas de um livro educacional de sistema operacional e, por final, apresentamos algumas estatísticas sobre a execução da ferramenta.

5.1 PROTÓTIPO

O protótipo foi implementado na linguagem Java, e está dividido em quatro pacotes principais (como pode ser visto na Figura 5.1): qa.analysis, qa.database, qa.extractanswer e qa.generation. Cada um dos pacotes possui objetivos bem definidos, que estão alinhados com as características conceituais apresentadas na Seção 3.1.

O sistema foi desenvolvido para trabalhar exclusivamente com perguntas da língua inglesa e, portanto sua codificação e nomenclatura também foram desenvolvidas em inglês.

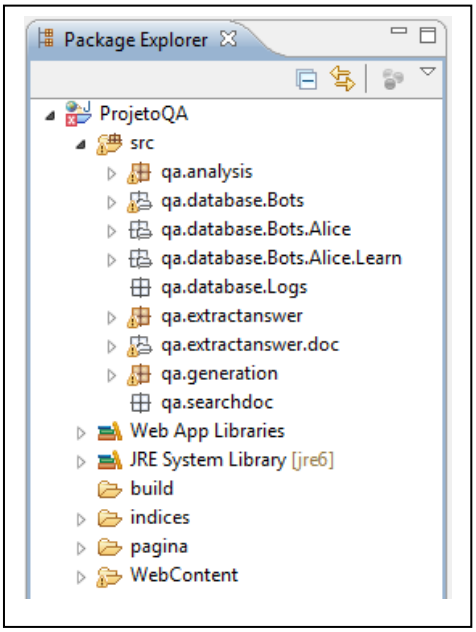

Figura 5.1 – Pacote Java do projeto

Para clarear o funcionamento interno do projeto, simulamos com um exemplo a execução do sistema. Para o parâmetro de entrada, selecionamos a pergunta: "*What are the basic functions of an operating system*?". A Figura 5.2 ilustra o funcionamento completo do sistema.

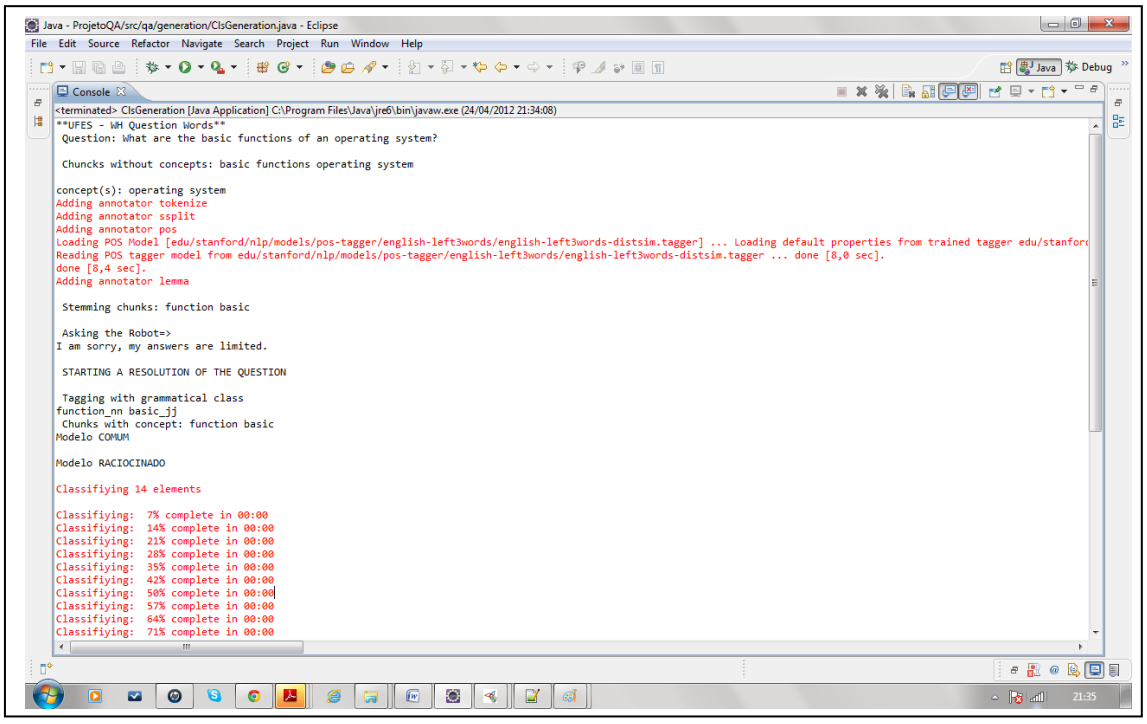

Figura 5.2 – Exemplo da tela de execução do sistema

O primeiro passo na execução é eliminar as *stopwords*, que são neste caso: *what*, *are*, *the*, *of* e *an*. A Figura 5.3 ilustra o funcionamento. A biblioteca Lucene elimina as stopwords.

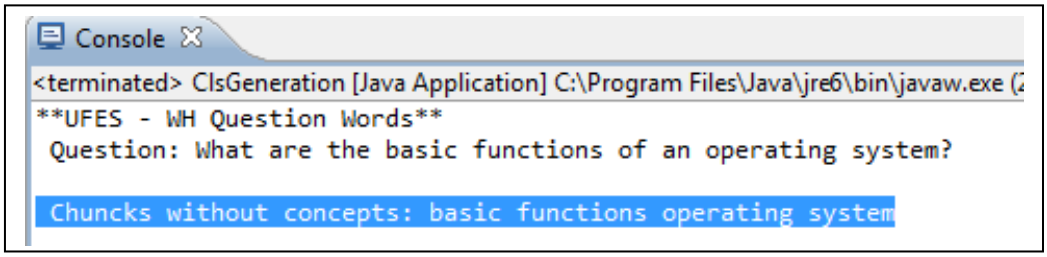

Figura 5.3 – Eliminando as *stopwords*

O próximo passo é selecionar e identificar os conceitos pertencentes ao domínio de sistema operacional. Como visto na Figura 5.4, o conceito *operating system* é selecionado.

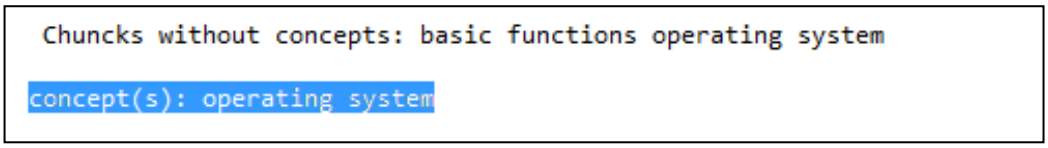

Figura 5.4 – Seleção dos conceitos

Após a seleção dos conceitos, as palavras restantes da pergunta são lematizadas. A lematização é realizada pela biblioteca Stanford NLP. A Figura 5.5 ilustra o passo:

Figura 5.5 – Lematização das palavras

Todos os passos anteriores se configuram como parte da análise da pergunta. Esse passo é necessário para gerar uma consulta que irá extrair a resposta. De posse da pergunta analisada o sistema deve consultar a base AIML. Caso encontre a resposta então o sistema deve retornar o resultado, conforme ilustrado na Figura 5.6. Caso não encontre a resposta o agente responsável pela análise e extração da pergunta inicia a resolução da resposta.

```
-

 Asking the Robot=>
The function basic are: Operating system controls and coordinates the use of the hardware among the various applications
```

Figura 5.6 – Resposta retornada da base AIML

Para auxiliar na extração da resposta o agente inicia a chamada do módulo de análise da pergunta, que irá agregar sinônimos aos substantivos da pergunta. Antes de agregar os sinônimos, é realizado um reconhecimento das classes gramaticais da palavra, como pode ser visto na Figura 5.7. A palavra que possui a etiqueta "_nn" é reconhecida como substantivo, em nosso exemplo foi identificado à palavra "function_nn". Essa ação de reconhecimento das classes gramaticais é realizada pelo Lingpipe.

```
Console
<terminated> ClsGeneration [Java Application] C:\Program

 STARTING A RESOLUTION OF THE QUESTION

 Tagging with grammatical class
function_nn basic_jj
```

Figura 5.7 – Reconhecimento das classes gramaticais

Em seguida o agente consulta na *Wordnet* os sinônimos do substantivo "*function*" e os agrega na pergunta analisada. A Figura 5.8 ilustra os sinônimos retornados.

```
Reading the meaning on wordnet for function and querying the synonyms.
function purpose role use

 Attaching the synonyms in question.
OS function basic purpose role use operating system
```

Figura 5.8 – Lendo os significados da *Wordnet* e os anexado a pergunta analisada

O próximo passo é agregar os conceitos da ontologia a fim de enriquecer a pergunta analisada. Portanto o conceito "*operating system*" é submetido à ontologia. O resultado da inferência é ilustrado na Figura 5.8. Nesse exemplo a inferência na ontologia retornou o conceito "OS" que é fruto de uma relação lógica do tipo equivalente.

Nesse passo o tipo semântico da pergunta é retornado, conforme ilustrado na Figura 5.9. Para resolver o tipo semântico o sistema reconhece padrões na pergunta com o auxílio de expressões regulares.

```
 Find out the semantic type - WH Question Words
Entity target: [DEFINITION]
Semantic type: FACTOID
```

Figura 5.9 – Tipo semântico identificado

A partir da pergunta analisada o sistema seleciona as páginas web que, por sua vez, passam por um processo de limpeza, ou seja, a formatação HTML é removida e o texto é extraído. O texto extraído das páginas é processado pela biblioteca JIRS que extrai as respostas prováveis para a pergunta. A figura 5.10 ilustra as respostas prováveis:

```
Run summary:
 Finish time of run execution (finishrun): 25/04/2012 21:59:53Load indexes data into memory:
177     0.95607704      pagina3.txt     It is the first program loaded into the computer by
0       0.9514436       pagina1.txt     What are the basic functions of an operating system
107     0.92532766      pagina2.txt     Time-sharing operating systems schedule tasks for ef
```

Figura 5.10 – Respostas prováveis

O sistema seleciona entre as respostas prováveis a de maior probabilidade, que passa por uma confirmação para verificar se é uma resposta válida. Esse processo de confirmação é realizado por um serviço executado pela biblioteca VENSES. Após o processo de confirmação, a resposta é formatada de acordo com o tipo semântico esperado. A Figura 5.11 mostra a resposta final.

```
Answer-> basic functions of an operating system are: Booting the computer
Performs basic computer tasks eg managing the various peripheral devices eg
mouse , keyboard Provides a user interface , e.g .command line , graphical user
interface (GUI) Handles system resources such as computer's memory and sharing
of the central processing unit (CPU) time by various applications or peripheral
devices Provides file management which refers to the way that the operating
system manipulates , stores , retrieves and saves data .
```

Figura 5.11 – Resposta retornada

5.2 EXPERIMENTO

Nesta seção apresentamos todo processo realizado no experimento com 60 perguntas selecionadas de livros educacionais de sistemas operacionais (MACHADO *et al.*, 2008) (TANENBAUM *et al.*, 2006). No experimento geramos dados quantitativos úteis para a avaliação da proposta do sistema.

5.2.1 Métricas de Avaliação

A avaliação será baseada em métricas comumente utilizadas em sistemas de pergunta-resposta, que são típicas em sistemas de recuperação de informação. As métricas são:

- Precisão (*Precision*) – é o número de perguntas respondidas corretamente divididas pelo número de perguntas respondidas (CIMIANO *et al.*, 2007).

- Abrangência (*Recall*) – é o número de perguntas respondidas pelo sistema dividido pelo número total de perguntas (POPESCU *et al.*, 2003).

5.2.2 Resultados

Os resultados dos experimentos são registrados na Tabela 5.2.

Tabela 5.1 – Quantidade de perguntas testadas

Descrição	**Quantidade**	**Percentual**
Quantidade total de perguntas	60	100%
Quantidade de perguntas respondidas com Ontologia e RTE	45	75%
Quantidade de perguntas respondidas sem RTE	49	81,67%
Quantidade de perguntas respondidas sem ontologia	57	95%
Quantidade de perguntas respondidas **corretamente** com Ontologia e RTE	43	71,67%

Descrição	**Quantidade**	**Percentual**
Quantidade de perguntas respondidas **corretamente** sem RTE	35	58,33%
Quantidade de perguntas respondidas **corretamente** sem Ontologia	26	41,67%

Os resultados de *recall* e *precision* são registrados na Tabela 5.3 e podemos perceber que o uso das ontologias e RTE implicam na melhora da qualidade do que foi recuperado (*precision*), entretanto há uma redução na quantidade das respostas recuperadas (*recall*).

Tabela 5.2 – *Precision* e *Recall*

Descrição	**Valor**
Precision com Ontologia e RTE	0,955
Recall com Ontologia e RTE	0,75
Precision sem Ontologia	0,438
Recall sem Ontologia	0,95
Precision sem RTE	0,714
Recall sem RTE	0,816

O Gráfico 5.12 compara a medida *recall* com o uso de ontologias e RTE no retorno das respostas.

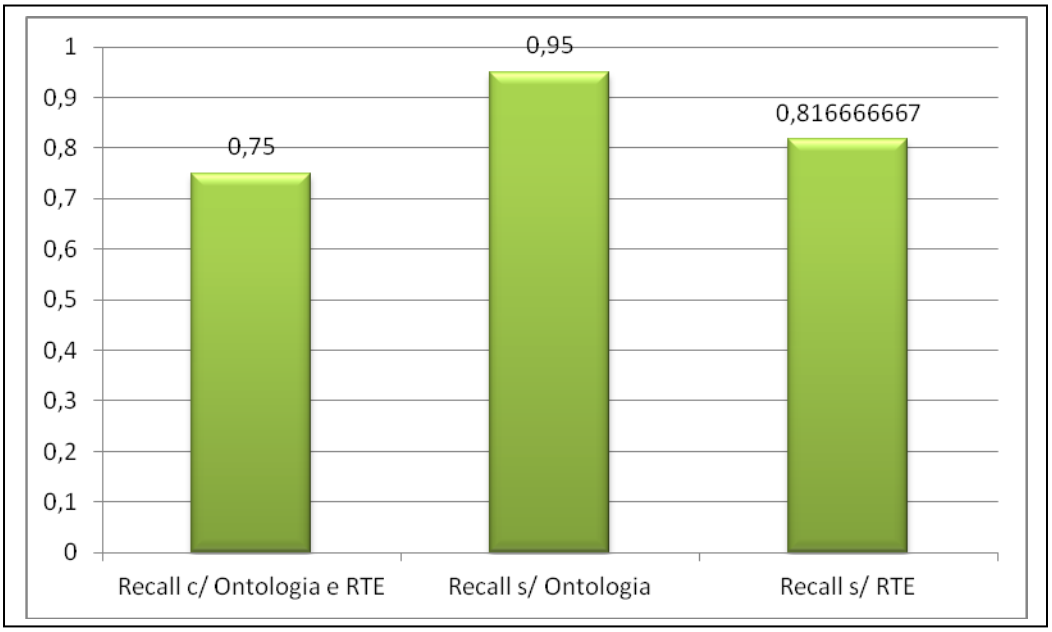

Figura 5.12 – Gráfico do Cálculo do *Recall*

O Gráfico 5.13 compara a medida *precision* com o uso de ontologias e RTE no retorno das respostas.

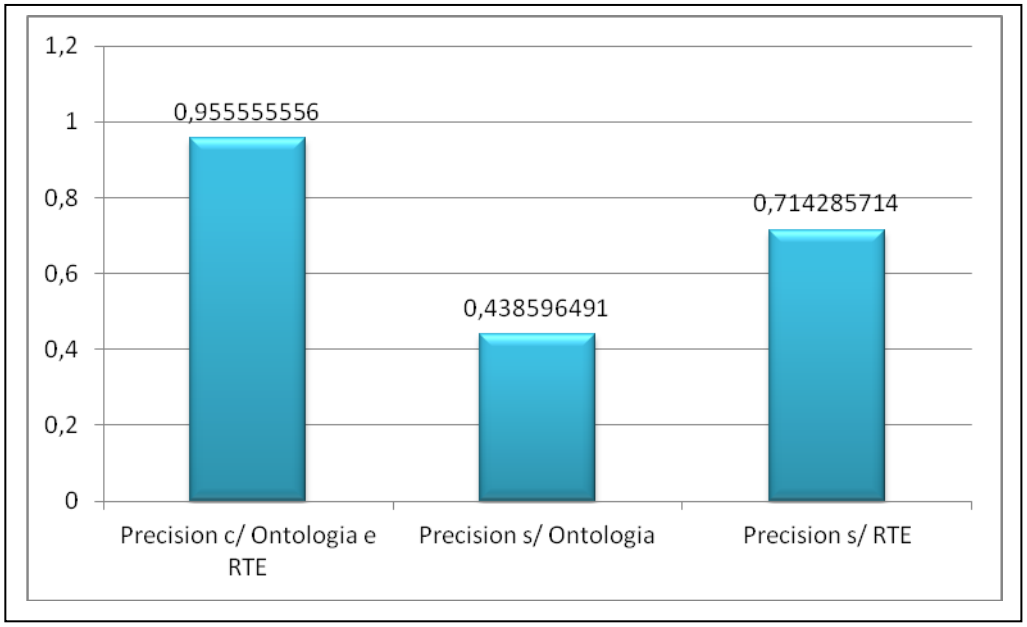

Figura 5.13 – Gráfico do Cálculo da *Precision*

- Para que ontologias aperfeiçoem sistemas de pergunta-resposta, é necessário estruturar os conceitos adequadamente. Por exemplo, a pergunta: "*what are*

types the operating systems?" por abranger outros domínios poderia gerar as respostas válidas: "32 bits", "64 bits" ou "mobile", no entanto para o domínio de conceitos básicos de sistema operacionais a resposta deveria ser "*batch*, *multitasking*, *monotasking*". Para melhorar a extração da resposta, novos conceitos devem ser buscados na ontologia, tal como "*multiprogramming*". Buscar os conceitos corretos é importante para expandir à pergunta para o domínio pretendido. Por isso que o uso de ontologias melhora na qualidade da medida de precisão (*precision*).

- O aumento do *recall* está associado aos conceitos presentes na ontologia. Quanto mais o domínio for descrito na ontologia, melhor será o *recall*. Para trabalhos futuros pretendemos ampliar a ontologia construída neste trabalho.

- A adição de novos conceitos da ontologia na pergunta analisada é muito útil para realizar buscas pela Web, mas se torna inútil na extração das passagens. A ocorrência de conceitos não existentes nas passagens reduz o valor da similaridade.

- A Web é composta de diversas mídias como pdf, *powerpoint*, entre outras, a possibilidade de pesquisar nessas fontes pode trazer vantagens na recuperação. O sistema proposto não tratou dessas fontes, mas podemos perceber que muitos desses documentos continham as respostas para as perguntas.

- As perguntas que possuem as respostas na base AIML retornam em um tempo bem menor do que as perguntas que passam pelo processo de resolução (análise, extração etc).

Os resultados confirmam a proposta do trabalho, demonstrando que o uso de ontologias, RTE e uma base de AIML são importantes na recuperação de sistemas de pergunta-resposta.

CAPÍTULO 6 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho desenvolveu um sistema de pergunta-resposta objetivando perguntas do tipo *WH-question* da língua inglesa. No desenvolvimento do projeto foram adicionadas ontologias e diversas outras técnicas úteis para aperfeiçoar a recuperação da resposta. Os sistemas atuais de pergunta-resposta utilizam muitas dessas técnicas, mas poucos utilizam todas em conjunto.

Através dos experimentos realizados em perguntas do domínio de sistemas operacionais, foi constatado que a proposta de utilização de ontologia, RTE e base conhecimento AIML são viáveis e importantes para melhorar a recuperação do sistema. Entretanto os especialistas do domínio ainda são importantes para avaliar as entradas (pergunta/resposta) na base de conhecimento.

Dentre os métodos introduzidos nessa arquitetura destacamos a transformação da pergunta em uma consulta apta para extrair os trechos relevantes.

Dentre as soluções tecnológicas disponíveis para o desenvolvimento desse sistema, o componente (JIRS) de extração das respostas prováveis. O algoritmo do componente se mostrou extremamente eficiente extraindo frases completas e candidatas a responder as perguntas.

A construção de um sistema de pergunta-resposta permite navegar em várias áreas de conhecimento (recuperação de informação, processamento de linguagem natural, ontologias, entre outras). Podemos concluir deste trabalho que a fusão dessas áreas é igualmente importante para evolução dos sistemas de pergunta-resposta.

Para os trabalhos futuros do sistema proposto citamos:

- Realizar experimentos reais, ou seja, criar uma base com perguntas e permitir que o sistema interaja com grupos de usuários.

- O sistema deve enviar para os especialistas as perguntas não resolvidas e resolvidas. Para as perguntas resolvidas o especialista deve avaliar o grau de qualidade da resposta. Para as perguntas não resolvidas os especialistas devem responder e solicitar que o sistema classifique e armazena na base AIML. Se uma pergunta receber várias respostas, o sistema deve identificar as diferenças léxicas e semânticas, e a partir disso armazenar as respostas no banco com um percentual de importância.

- É necessário um mecanismo para enriquecer a ontologia com novos conceitos, relações e instâncias do domínio a partir de textos da Web. A inclusão de novos conceitos deve contar com o auxílio dos especialistas.

- Ampliar a comunicação do sistema de pergunta-resposta com outros sistemas através da troca semântica de dados. Dessa maneira outros sistemas poderiam consumir o serviço de pergunta-resposta sem precisar conhecer o funcionamento interno do sistema. Por exemplo: ambientes de aprendizagem virtuais poderiam utilizar o método de pergunta-resposta para testar o conhecimento do aluno.

- Permitir que o sistema interaja com múltiplos idiomas, ou seja, receber a pergunta em português ou inglês e retornar a resposta de acordo com o idioma do usuário.

- Resolução de anáforas. A resolução de perguntas que se referem a outras perguntas que já ocorreram anteriormente.

- Trabalhar com a fusão de respostas de diferentes fontes de documentos. Se partes da resposta estão em documentos diferentes, criar um algoritmo para unir essas partes em um única resposta.

- Expandir os tipos de perguntas resolvidas pelo sistema, ou seja, além das *Wh-question* permitir a resolução de perguntas de outro tipo semântico (por exemplo: raciocínio (sim/não).

- Transformar os módulos do sistema (análise da pergunta, gerador da resposta, seleção e extração da resposta) em agentes inteligentes, como pode ser visto na Figura 6.1. A utilização de agentes possibilita criar uma arquitetura mais dinâmica, em que cada base de conhecimento poderia ser alimentada sempre que novas informações ou conceitos fossem encontrados. Além disso, pode-se trabalhar de forma independente, ou seja, enquanto uma parte dos agentes está resolvendo a pergunta, outra parte pode atualizar as bases de conhecimento.

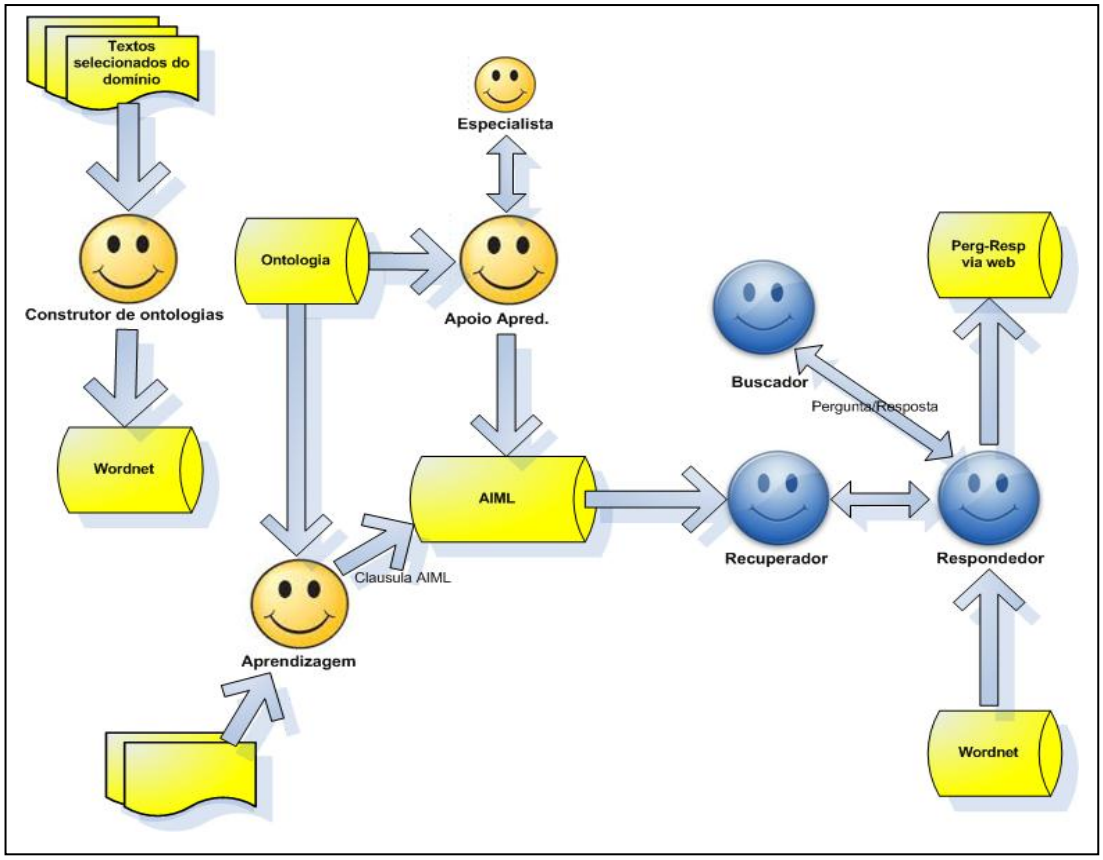

Figura 6.1 – Arquitetura proposta para trabalhos futuros

REFERÊNCIAS

AIML, "**Artificial Intelligence Markup Language**", Disponível em: http://www.alicebot.org/aiml.html . Acesso em: 11 de Maio de 2012.

ABRAHAM, A.; MAURI, L. J.; BUFORD, J. F.; SUZUKI, J. THAMPI, S. M.; **Advances in computing and communications**, Part III: First International Conference, Spring-Verlag, pag. 146-153, 2011.

AKERKAR, R. A.; SAJJA P. S. **Knowledge-Based system**, capítulo Natural Language Interface: Question Answering System, pag. 323-330. Jones and Barlett Publishers, 2010.

ATHENIKOS, S. J.; HYOIL, H. **Biomedical question answering: A survey**, Journal Computer Methods and Programs in Biomedicine, Volume 99, Issue 1, Elsevier, New York, USA, 2010.

BAEZA-YATES, R.; RIBEIRO-NETO, B. **Modernal information retrieval**, Addison Wesley, USA, 2000.

BENTIVOGLI, L.; DAGAN, I; DAGAN, H. T.; GIAMPICCOLO, D.; MAGNINI, B. **The Fifth PASCAL recognizing textual entailment challenge**, In Proceedings of TAC 2008 workshop, 2009.

BOSCH, A.V. D. ; BOUMA, G. **Interactive multi-modal question-answering**, Springer-Verlag, 2011.

BOUMA, G.; FAHMI, I.; MUR, J. **Interactive multi-modal question-answering**, capítulo Relation Extraction for Open and Closed Domain Question Answering, pag. 171-190. Springer-Verlag, 2011.

BROOKSHEAR, G. J. **Computer Science: An Overview,** Publisher Addison Wesley, 11 edition, 2011.

BUSCALDI, D.; ROSSO, P.; GÓMEZ-SORIANO, J. M.; SANCHIS, E. **Answering question with an n-gram based passage retrievak engine**, In Journal of intelligent information systems, Volume 34, Issue 2, Kluwer Academic Publishers, Maime, USA, 2010.

BURHANS, D. T. **A question-answering interpretation of resolution refutation**, capítulo Introduction: Question Answering, Doctoral Dissertation, pag. 1-2. University at Buffalo, New York, USA, 2002.

CAO, Y.; LIU, F.; SIMPSON, P.; ANTIEAU, L.; BENNETT, A.; CIMINO, J. J.; ELY, J.; YU, H. **AskHERMES: An online question answering system for complex clinical question**, Journal of Biomedical Informatics, Volume 44, Issue 2, Publisher Elsevier Science, San Diego, USA, 2011.

CARPINETO, C.; ROMANO, G. **A survey of automatic query expansion in information retrieval**, Journal ACM Computing Surveys, Volume 44, Issue 1, New York, USA, 2012.

CHIRITA, A-P.; FIRAN, S. C.; NEJDAL, W. **Personalized query expasion for the web**, In Proceedings SIGIR'07 proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval, pag. 7-14, New York, USA, 2007.

CHURCH, K.; PATIL, R. **Coping with syntatic ambiguity or how to put the block in the box on the table**, Journal Computational Linguistics, MIT Press Cambridge, Volume 8, pag. 139-149, 1982.

CIMIANO, P.; HAASE, P.; HEIZMANN, J. **Porting natural language interfaces between domains: an experimental user study with the ORAKEL system**, In Proceedings of the 12th International Conference on Intelligent User Interfaces, pag. 180-189, New York USA, 2007.

CLARK, A.; FOX, C.; LAPPIN S. **The: Handbook of computacional linguistics and natural language processing**, capítulo Question and Answering, pag. 630-654. Wiley-Blackwell, 2010.

DAGAN, I.; DOLAN, B; MAGNINI, B.; ROTH D. **Recognizing textual entailment: Rational evaluation and approaches**, Natural Language Engineering, Cambridge Universuty Press, 2009.

DAMIJANOVIC, D.; AGATONOVIC, M.; CUNNINGHAM, H.; **Natural language interfaces to ontologies: combining syntactic analysus and ontology-based lookup through the user interaction**, In Proceedings of the 7th extended semantic web conference, Crete, Greece, pag. 106-120, 2010.

FALBO, R. A.; GUIZZARDI, G.; DUARTE K. C. **An Ontological approach to domain engineering**, In Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering, 2002.

FENSEL, D. **Problem-Solving methods**, capítulo A Four Component Architecture for Knowledge-Based Systems, pag. 43-46. Springer-Verlag, 2000.

FERRÁNDEZ, O.; SPURK, C.; KOUYLEKOV, M.; DORNESCU, I; FERRÁDEZ, S.; NEGRI, M.; IZQUIERDO, R.; TOMÁS, D.; ORASAN, C.; NEUMANN, G.; MAGNINI, B.; VICEDO, J. L. **The QUALL-ME Framework: A Specifiable-Domain**

multilingual question answering architecture, capítulo Question Answering via Web Services Composition, pag. 29-37. Springer Berlin Heidelberg, 2011.

FERRUCCI, D.; NYBERG, E.; ALLAN, J.; BARKERE, K.; BROWN, E.; CHU-CARROL, J.; CICCOLO, A.; DUBOUE, P.; FAN, J.; GONDEK, D.; HOVY, E.; KATZ, B.; LALLY, A.; MCCORD, M.; MORARESCU, P.; MURDOCK, B.; PORTER, B.; PRAGER, J.; STRZALKOWSKI, T.; WELTY, C.; ZADROZNY, W. **Toward the open advancement of question answering systems**, ICM Research Division, Thomas J. Watson Research Center, 2009.

FLIEDNER, G. **Linguistically informed question answering**, capítulo Question Answering and Other Information Access Systems, pag. 7-32. 2007.

GABBAY, D. M.; JOHNSON, R. H.; OHLBACH, H. J.; WOODS, J. **Handbook of logic of argument inference**, Elsevier Science, 1 edition, Amsterdam, The Netherlands, 2002.

GALVÃO, A. M.; BARROS, F. A.; NEVES, A. M. M.; RAMALHO, L. G. **Persona-AIML: An Architecture for Developing**, Chatterbots with personality, Universidade Federal de Pernambuco, 2004.

GIL A. C. **Como Elaborar Projetos de Pesquisa**, capítulo Como Encaminhar uma Pesquisa, pag. 17-22, Editora Atlas, 2002.

GLÖCKNER, I.; PELZER, B. **Extending a logic-based question answering system for administrative texts**, In Proceedings of the 10th cross-language evaluation forum conference on Multilingual information access evaluation: text retrieval experiments, Springer-Verlag Berlin, pag. 265-272, 2009.

GÓMEZ-PÉREZ, A.; FERNÁNDEZ-LÓPEZ, M.; CORCHO O. **Ontological engineering: with examples from areas of knowledge management, e-commerce and semantic Web**, Springer-Verlag, New York, USA, 2007.

GONZALEZ, M.; LIMA, L. S. de V. **Recuperação de Informação e processamento da linguagem natural**, PUCRS, Porto Alegre, Brasil, 2003.

GRAPPY, A.; GRAU, B.; FALCO, M-H.; LIGOZAT, A-L.; ROBBA, I.;VILNAT, A. **Selecting answers to questions from Web documents by a robust validation process**, In Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, Volume 01, pag. 55-62, 2011.

GREEN, B. F.; WOLF, A. K.; CHOMSKY, C.; LAUGHERY, K. **BASEBALL: An automatic question answerer**. In Proceedings Western Joint Computer Conference 19, pag. 219-224, New York, USA, 1961.

GRUNINGER, M.; FOX, M. S. **Methodology for the Design and Evaluation of Ontologies**, Technical Report, University of Toronto, 1995.

GUARINO N. **Formal Ontology in information systems**, In Proceedings of the first international conference (FOIS'98), Trento, Italia, 1998.

GUNNING, D.; CHAUDHRI, V. K.; WELTY, C. **Introduction to the special issue on question answering**, Association for the Advancement of Artificial Intelligence, 2010.

HATCHER, E.; GOSPODNETIC, O.; MCCANDLESS, M. **Lucene in Action**, Romania: Phd Thesis, Manning Publications Science, Second Edition, 2009.

HARABAGIU, S. M.; MAIORANO, S. J.; PASCA, M. A. **Dialogue management for interactive question answering**, In Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference, AAAI Press, 2001.

HIRSCHMAN L.; GAIZAUSKAS R. **Natural Language question answering: the view from here**, journal natural language engineering, Volume 7 Issue 4, 2001.

HORRIDGE, M.; BRANDT, S. **A Pratical Guide to Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3**, The University of Manchester, 2011.

IFTENE, A. **Textual Entailmente**, Romania: Phd Thesis, Computer Science, University of Iasi, 2009.

JONES, K. S.; WILLETT, P. **Readings in information retrieval**, Morgan Kaufmann Publishers, 1997.

JURCZYK, P.; AGICHTEIN, E. **Discovering authorities in question answer communities by using link analysis**, In Proceedings of the sixteenth ACM conference on information and knowledge management, New York, USA, pag. 912-922, 2007.

KIYOTA, Y.; KUROHASHI, S.; KIDO, F. **"Dialog Navigator" a question answering system based on large text knowledge**, In Proceedings COLING'02 Procedings of the international conference computational linguistics, Pennsylvania, USA, Volume 1, pag. 1-7, 2002.

KONCHADY, M. **Build sarch applications Lucene, LingPipe, Gate**, capítulo Future Directions in Search, pag. 369-392. Mustru Publishing, 2008.

KONOPÍK, M.; ROHLÍK, O. **Question Answering for not yet semantic web**, In Proceedings of 13th international conference text, speech and dialogue,Brno,Cze Republic, pag. 125-132, 2010.

LEI, G.; LI, G; ZHENG, Y-T; HONG, R.; CHUA, T-S. **Video reference: a video question answering engine**, In Proceeding of the 16th international conference on Advances in Multimedia Modeling,Spring-Verlag, Berlin, Heidelberg páginas 1-7, 2010.

LI, X.; ROTH, D. **Learning question classifiers**, In Proceedings of the 19th international conference on computational linguistics,Volume 1, páginas 1-7, 2002.

LIGÊZA, A. **Logical foundations for rule-based systems**, Springer-Verlag, Berlin, 2006.

LINGPIPE, "Tool kit for processing text using computational linguistics", Disponível em: http://alias-i.com/lingpipe/ . Acesso em: 11 de Maio de 2012.

LIU, H.; LIN, X.; LIU, C.; **Research and Implementation of Ontological QA System Based on FAQ**, In Journal of Convergence Information Technology, Volume 5, Número 3, páginas 79-85, 2010.

LOPEZ, V.; FERNÁNDEZ, M; STIELER, N.; MOTTA, E. **Discovering authorities in question answer communities by using link analysis**, In Journal Web Semantic, Disponível em: <http://www.semantic-web-journal.net>, 2011.

MACHADO, F. B.; MAIA, L. P. **Arquitetura de Sistema Operacionais**, Editora LCT, 4 edição, 2008.

MAYBURY, M. T. **New directions in question answering**, MIT Press, Stanford, USA, 2004.

MENG, W.; YU, C. **Advanced Metasearch Engine Technology**, Morgan & Claypool Publishers, 2010.

MINOCK, M. **Where are the 'Killer Applications' of Restricted Domain Question Answering**, In Proceedings of the IJCAI Workshop on Knowledge Reasoning in Question Answering, Edinburgh, Scotland, 2005.

MOLDOVAN, D.; HARABAGIU, S.; PASCA, M.; MIHALCEA, R.; GOODRUM, R.; GÎRJU, R.; RUS, V. **LASSO: A Tool for Surfing the answer Net**, In Proceeding of the Eighth Text Retrieval Conference (TREC-8), 1999.

MONZ, C. **Information retrieval to question answering**, PhD Thesis, University of Amsterdam, 2003.

MOSCHITTI, A.; QUARTERONI, S.; BASILI, R; MANDHAR, S. **Exploiting syntactic and shallow semantic kernels for question/answer classification**, In Proceedings ICTAI '08 proceedings of the 2008 IEEE international conference on tools with artificial intelligence, Washington, USA, pag. 123-130, 2008.

NAVIGLI, R.; FARALLI, S.; SOROA, A.; LACALLE, O.; AGIRRE, E. **Two birds with one stone: learning semantic models for text categorization and word sense disambigation**, In Proceedings of the 20th ACM international conference on Information and knowledge management, New York, USA, 2011.

NOY, N. F.; MCGUINNESS, D. L. **Ontology Development 101: A Guide to Creating your First Ontology**, Stanford University, 2001.

OH, H-J.; MYAENG, S. H.; JANG, M-G.; **Effects of answer weight boosting in strategy-driven question answering**, Information Processing and Management, Pergamon Press, pag. 83-93, Nova York, USA, 2012.

PLESSERS, P.; TROYER, O. **Resolving inconsistencies in evolving ontologies –**, In Proceedings of the 3rd European conference on The Semantic Web research and applications, pag. 200-214, Springer-Verlag, 2006.

POPESCU, A-M.; ETZIONI, O.; KAUTZ, H. **Toward a theory of natural language interfaces to databases –**, In Proceedings of the 8th internacional conference on intelligent user interfaces, pag. 149-157, New York, USA, 2003.

PRAGER, J.; BROWN, E.; CODEN, A. **Question-answering by predictive annotation**, In Proceedings SIGIR'00 proceedings of the 23th annual international ACM SIGIR conference on research and development in information retrieval, pag. 184-191, New York, USA, 2000.

STRZALKOWSKI, T.; HARABAGIU, S. **Advances in open domain question answering**, capítulo Coping with alternate formulations of questions and answer, pag. 189-226, Springer, 2008.

SIMMONS, R. F. **Answering english questions by computer: A survey**, Communications of the ACM, Volume 8, Issue 1, 1965.

SOMAN, K. P.; LOGANATHAN, R.; AJAY, V. **Machine learning with SVM and other kernel methods**, PHI Learning Private Limited, 2009.

STANFORD, "**The Stanford Natural Language Processing Group**", Disponível em: http://nlp.stanford.edu . Acesso em: 12 de Setembro de 2012.

TANENBAUM, A. S.; WOODHULL, A. S. **Operating systems: design and implementation**, Editora Pearson Prentice Hall, 3 edição, Universidade da Califórnia , 2006.

TELLEX, S.; KATZ, B.; LIN, J.; FERNANDES, A.; MARTON, G. **Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering**, In Proceedings of the 26th Annual Internation ACM SIGIR Conference, 2003.

TEIXEIRA S. **ChatterBots – Uma Proposta para a Construção de Bases de Conhecimento**, Dissertação de mestrado apresentado ao programa de pós-gradução em informática do centro técnologico, Universidade Federal do Espirito Santo, 2005.

TORRES-MORENO, J-M.; PIER-LUC, St-O.; GAGNON, M.; EL-BÈZE, M.; BELLOT, P. **Automatic Summarization System Coupled with a Question-Answering System (QAAS)**, NLP New. Computation and Language, 2009.

VARGAS, M. **Técnica, tecnologia e ciência**, Revista Educação & Tecnologia, 2011.

VENSES, "Venice Semantic Evaluation System", Disponível em: http://project.cgm.unive.it/venses_en.html. Acesso em: 11 de Maio de 2012.

VICEDO, J. L.; MOLLÁ, D. **Open-Domain Question-Answering State of the Art and Future Trends OU Question Answering in Restricted Domain: An Overview**, Journal Computational Linguistics, Volume 33, Issue 1, Mit Press, 2007.

WAHLSTER, W.; MARBURGER, H.; JAMESON, A.; BUSEMANN, S. **Over-answering yes-no questions: extended responses in a NL interface to a vision system**, In Proceedings of the eighth international joint conference on artificial intelligence, San Francisco, USA, pag. 643-646, 1983.

WANG, L.; LIAO, L.; WANG, X. **Question answering via semantic web service composition**, Advances in Control and Communication, Spring-Verlag, pag. 29-37, 2012.

WANG, D.; LI, T.; ZHU, S.; GONG, Y. **iHelp: An Intelligent Online Helpdesk System**, IEEE Transactions on systems, man, and cybernetics, 2011.

W3C, "**World Wide Web Consortium**", Disponível em: http://www.w3.org. Acesso em: 12 de Setembro de 2012.

ZHENG, Z. **AnswerBus Question Answering System**, In Proceeding of the second international conference on human language technology research, Morgan Kaufmann Publishers, San Francisco, USA, 2002.

ZOUAQ, A.; NKAMBOU, R.. **Evaluating the generation of domain ontologies in the knowledge puzzle project**, Journal IEEE Transactions on Knowledge and Data Engineering,Volume 21, pag. 1559-1572, 2009.

APÊNDICES

APÊNDICE A – QUESTÕES DE COMPETÊNCIA DA ONTOLOGIA

1. What are the basic functions of an operating system?
2. What are types the operating systems?
3. What is difference between uniprogramming and multiprogramming system?
4. What are the characteristics of a batch system?
5. How work time-sharing systems?
6. What is difference tightly coupled between loosely coupled systems?
7. What is SMP system?
8. What are the advantages of systems multiprogramming?
9. What is virtual machine?
10. What is tightly coupled systems?
11. What is loosely coupled systems?
12. What are the functional units of a computer?
13. Which components of a processor?
14. What are the functions of a processor?
15. How the main memory of a computer is organized?
16. What are volatile memory?
17. What are nonvolatile memory?
18. What are the functions of the linker?
19. What are the functions of the loader?
20. What is RISC processor?
21. What is CISC processor?
22. What is concurrency?
23. Why the interrupt mechanism is essential in multiprogramming?
24. What are synchronous events?
25. What are asynchronous events?
26. What is the kernel?
27. What are main function of kernel?
28. What are privileged instructions?
29. How to work the change the mode of access?
30. What is system call?
31. How does the client-server in microkernel architecture?
32. Why the routines of the operating system possess privileged instructions?

33. How does the activation process (boot) of the operating system?
34. What are monolithic architecture?
35. What is the process?
36. What parts make up a process?
37. What is function the software context?
38. What are the five possible states of a process?
39. What is the difference between multithreading and subprocesses?
40. What is the difference between background and foreground processes?
41. What is an address space of a process?
42. What is the data structure indicated to organize the processes in the main memory?
43. How the elimination of a process uses mechanisms signs?
44. What are examples of applications CPU-bound and IO-bound?
45. What is a multithreaded environment?
46. How an application can implement competition in an monothread environment?
47. What is the difference between unit resource allocation and scheduling unit?
48. What is the advantage of sharing the address space between threads of the same process?
49. What are the problems of competing applications developed in environments monothreads?
50. What is a monothread environment?
51. What are the advantages of a monothread environment?
52. What is the difference between resource allocation unit and scheduling unit?
53. What is advantage of Scheduler Activations compared to the hybrid?
54. What are the benefits of using threads in a client-server environment?
55. How the use of threads can be useful in microkernel architecture?
56. What is not an advantage of distributed systems?
57. What is Throughput?
58. How can be implemented the virtual memory ?
59. What resources are used when a thread created?
60. What is fragmentation?

GLOSSÁRIO

Lista dos termos técnicos, siglas, jargões e estrangeirismo seguido da definição.

1) Inferência lógica – Existem várias formas de raciocínio. Pessoas aplicam mais de 20 formas diferentes de raciocíonio. O raciocínio pode ser preciso ou vago, probabilístico, fuzzy, etc. Na lógica clássica os paradigmas de raciocíonio são: dedução, abdução e indução. A dedução é o método mais conhecido da inferência lógica. A inferência lógica é realizada pela geração de novas sentenças ou fórmulas a partir de um conjunto inicial de hipóteses e com o uso de um conjunto específico de regras de inferências. Um exemplo de regra de inferência é o *modus ponens* (LIGÊZA, 2006).

2) Implicação textual (RTE) – Embora seja comumente usado o termo inferência textual, a terminologia usada neste trabalho é implicação textual. A implicação textual consiste na tarefa de desenvolver um sistema que, dado dois fragmentos de texto, é possível determinar se o significado de um texto é compreendido – ou seja, pode ser inferido – de outro texto. O RTE tem desfrutado de uma popularidade cada vez maior na comunidade de processamento de linguagem natural, entretanto existem outras técnicas (BENTIVOGLI *et al.*, 2009).

3) Algoritmo tableaux – O algoritmo permite checar a consistência da ontologia. O princípio básico do algoritmo usado para checar a satisfabilidade do conceito C é construir gradualmente um modelo I na qual C faz parte e não é vazio. O algoritmo constrói uma árvore, em que cada um nó da árvore corresponde a elementos da ontologia (PLESSERS *et al.*, 2006).

4) *POS Tagging* – Um *Part-Of-Speech Tagger* (*Tagging*) é um software que lê um texto em algum idioma e atribui partes da linguagem para cada palavra, por exemplo como: substantivo, verbo, adjetivo, entre outras (STANFORD, 2012).

5) CSS – *Cascading Style Sheets* é um simples mecanismo de adicionar estilo (exemplo:fontes, cores, espçamento) nas páginas Web (W3C, 2012).

6) SPARQL – É uma linguagem de consulta para RDF (W3C, 2012)..

7) RDF – É um modelo padrão para troca de dados na WEB (W3C, 2012).

8) *Tokens* - A menor unidade significativa de informação de uma sequência de dados (Oxford dictionary).