UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CIÊNCIA DA COMPUTAÇÃO

KASSIUS VARGAS PRESTES

Avaliação de métodos de Seleção da Resposta em um sistema de Perguntas e Respostas

Projeto de Diplomação

Profa. Dra. Aline Villavicencio
Orientador

Rodrigo Wilkens
Co-orientador

Porto Alegre, dezembro de 2011

“Luke: I don’t, I don’t believe it.
Yoda: That is why you fail. ”

AGRADECIMENTOS

Agradeço aos meus pais e à minha irmã por sempre me apoiarem em todas as etapas da minha vida.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS 6

LISTA DE FIGURAS 7

LISTA DE TABELAS 8

RESUMO 9

ABSTRACT 10

1 INTRODUÇÃO 11

2 TRABALHOS RELACIONADOS 13
2.1 *Question Answering* 13
2.1.1 Análise da Pergunta 13
2.1.2 Busca dos Documentos Candidatos 15
2.1.3 Geração das Respostas Candidatas 16
2.1.4 Seleção da Resposta 16
2.2 Linhas de Pesquisa em Question Answering 16
2.3 Principais Trabalhos 18
2.3.1 QA inglês 18
2.3.2 QA português 24
2.4 Seleção da Resposta 26
2.4.1 *Framework* probabilístico para seleção da resposta 26
2.4.2 Validação lógica 27
2.4.3 Sistema de Perguntas e Respostas orientado a estratégia 27
2.4.4 Seleção da Resposta utilizando SVM 28
2.5 Avaliação 28
2.5.1 TREC e CLEF 29
2.5.2 Avaliando sistemas de Perguntas e Respostas usando inserção de respostas de FAQs 29
2.6 Recursos 31
2.6.1 Ontologias 31
2.6.2 Parsers 33
2.6.3 Reconhecedor de Entidades Nomeadas 33

3 MATERIAIS E MÉTODOS 34
3.1 Corpus 34
3.2 Ferramentas 36
3.2.1 Stanford NER 36
3.2.2 Palavras 36
3.2.3 RASP Parser 37

4 ARQUITETURA 38
4.1 Sistemas Desenvolvidos 38
4.1.1 Sistema de Perguntas e Respostas para o português 38
4.1.2 Sistema de Perguntas e Respostas para o inglês 39
4.1.3 Sistema Estendido de Perguntas e Respostas para o inglês 40
4.2 Algoritmos de Seleção da Resposta 40
4.2.1 Similaridade *Bag-of-Words* 41
4.2.2 N-Grams 41
4.2.3 Distância *Keywords* 41
4.2.4 Google Score 42
4.2.5 Tf-Idf 42
4.2.6 Wikipedia Score 43
4.2.7 Algoritmos de Aprendizado de Máquina 43
4.2.8 Agrupamento de Respostas 44

5 AVALIAÇÃO 45
5.1 Primeira Etapa: Extração de Respostas 45
5.2 Segunda Etapa: Seleção da Resposta 45
5.2.1 Taxa de acerto para cada sistema 46
5.2.2 Comparação por Idioma 47
5.2.3 Comparação por recursos utilizados no sistema 48
5.2.4 Resultados dos sistemas completos 48
5.3 Análise dos resultados 49

6 CONCLUSÃO 50

REFERÊNCIAS 52

LISTA DE ABREVIATURAS E SIGLAS

QA	Question Answering
PLN	Processamento de Linguagem Natural
TREC	Text Retrieval Conference
CLEF	Cross Language Evaluation Forum
QA@CLEF	Trilha de Question Answering do CLEF
WEKA	Waikato Environment for Knowledge Analysis
PoS	Part-of-Speech
NP	Noun Phrase
FAQ	Frequently Asked Questions

LISTA DE FIGURAS

Figura 2.1: *Pipeline* padrão de um sistema de QA 13
Figura 2.2: A análise da pergunta no pipeline de um QA 14
Figura 2.3: A busca dos documentos candidatos no *pipeline* de um QA 15
Figura 2.4: A extração da Resposta no *pipeline* de um QA 16
Figura 2.5: A seleção da resposta no *pipeline* de um QA 17
Figura 2.6: Arquitetura do MULDER 19
Figura 2.7: Arquitetura do OpenEphyra 23
Figura 2.8: Identificação de um componente com erro usando inserção de respostas de FAQ 30
Figura 2.9: Ontologia de domínio do Comunica 31
Figura 2.10: A navegação na hierarquia de hiperônimos da WordNet 32

LISTA DE TABELAS

Tabela 2.1: Tipos de resposta esperados 14

Tabela 3.1: Distribuição das perguntas – português 34
Tabela 3.2: Distribuição das perguntas – inglês 34
Tabela 3.3: Tipos de Resposta – português 35
Tabela 3.4: Tipos de Resposta – inglês 35
Tabela 3.5: Questões válidas 35

Tabela 4.1: Regras utilizadas para o tipos de resposta esperado no português . . . 39
Tabela 4.2: Regras utilizadas para o tipos de resposta esperado no inglês 40

Tabela 5.1: Resultados da etapa 1 do experimento: a geração das respostas candidatas 45
Tabela 5.2: Resultados etapa 2 – seleção da resposta 46
Tabela 5.3: Resultados etapa 2 – Agrupamento 46
Tabela 5.4: Resultados dos algoritmos de aprendizado de máquina – português . . 47
Tabela 5.5: Resultados dos algoritmos de aprendizado de máquina – inglês 47
Tabela 5.6: Resultados português x Resultados inglês 47
Tabela 5.7: Resultados sistema para inglês x Resultados OpenEphyra 48
Tabela 5.8: Resultados dos sistemas completos 48

RESUMO

Este trabalho tem como objetivo a comparação de métodos de seleção da resposta entre três sistemas de Perguntas e Respostas (Question Answering), um sistema para o português, um sistema para o inglês e um sistema estendido para o inglês. Com essa comparação queremos investigar se os resultados de um sistema de Perguntas e Respostas para o português pode ter resultados tão bons quanto um sistema para o inglês. O foco desse trabalho será a avaliação de diferentes algoritmos utilizados na última etapa de um sistema de Perguntas e Respostas, a seleção da resposta.

Palavras-chave: Processamento de linguagem natural, question answering.

Evaluating answer selection methods in a Question Answering System

ABSTRACT

This paper aims at the comparison of methods of answer selection in three Question Answering Systems, one system for Portuguese, one system for english and one complex system for English. With this comparison we want to investigate if the results of a Question Answering system for Portuguese may have results as good as a system for English. This paper focuses in the evaluation of different algorithms used in the last stage of a Question Answering system, the answer selection.

Keywords: natural language processing, question answering.

1 INTRODUÇÃO

Atualmente há muita informação disponível na web para acesso de todos, porém essa informação é muito desorganizada, de forma que pode ser difícil encontrar algum tipo de informação específica. Uma das soluções para este problema da falta de organização da web são as ferramentas de busca existentes, como Google e Yahoo, que indexam o conteúdo disponível na web e ajudam as pessoas a encontrar informações relevantes sobre o que elas desejam. Estes buscadores são muito eficientes para determinados tipos de consultas, como por palavras chave, porém caso se esteja buscando apenas uma resposta para uma pergunta simples, a prioridade pode não ser encontradar milhões de páginas em menos de meio segundo, mas ao invés retornar somente a resposta da pergunta mesmo que para isso precise um pouco mais de tempo (SALLOUM, 2009).

Para resolver estas limitações dos sistemas de busca atuais de não conseguir responder a simples perguntas feitas em linguagem natural, como por exemplo, *"Quem é o presidente do Brasil?"*, em inglês *"Who is the president of Brazil?"* , cuja resposta é "Dilma Roussef", existem os sistemas de Perguntas e Respostas (em inglês, Question Answering, abreviado em QA).

Em Processamento de Linguagem Natural (PLN), um sistema de Perguntas e Respostas tem a tarefa de responder automaticamente uma pergunta em linguagem natural, procurando por informação em uma determinada fonte de dados, como uma base de dados estruturada ou documentos não estruturados em linguagem natural como jornais ou a web.

Os sistemas de QA se dividem em sistemas de domínio específico, que lidam com uma área específica e tem bons resultados pelo uso de ontologias de domínio que conseguem formalizar o conhecimento da área, como por exemplo o Comunica (WILKENS et al., 2010). Há também os sistemas de domínio aberto que respondem a questões sobre qualquer assunto, como o AnswerBus (ZHENG, 2002), o Javelin (NYBERG et al., 2002) e o MULDER (KWOK; ETZIONI; WELD, 2001), o que é uma tarefa mais difícil comparada à anterior, esse tipo de sistema utiliza ontologias gerais e tem como vantagem ter mais dados de onde extrair as respostas.

Um sistema de QA normalmente possui quatro estágios: (1) análise da pergunta, (2) busca dos documentos candidatos, (3) geração das respostas candidatas e (4) seleção da resposta. No estágio de análise da pergunta é identificado o tipo esperado de resposta (por exemplo, uma pessoa, data ou local) e as palavras chave, podendo ser realizado também identificação de entidades presentes na pergunta. Na busca dos documentos candidatos é enviada uma consulta a uma ferramenta de busca que retorna uma lista com os documentos mais prováveis de conterem a resposta para a consulta feita. Na geração das respostas candidatas são extraídas as partes desses documentos que contém possíveis respostas utilizando o tipo de resposta esperado e as palavras chave obtidas no estágio de análise da

questão. Por fim na seleção da resposta é calculado um score para cada resposta possível e criada uma lista ordenada com as respostas para a pergunta feita. A resposta com o maior score é escolhida como a resposta certa e é exibida para o usuário.

O desenvolvimento de sistemas de QA eficientes é um dos grandes desafios da área (BURGER et al., 2001) e há diversos trabalhos estudando as mais diversas técnicas e abordagens para sistemas de QA em diferentes línguas, porém a maioria destes artigos tem como língua alvo o inglês. Já para o português, os estudos neste assunto estão bem mais atrasados comparados ao inglês, tanto pela falta de recursos quanto de ferramentas adequadas.

Particularmente, a seleção da resposta é uma etapa muito importante e é um grande desafio, pois temos a tarefa de identificar a resposta correta entre uma lista de respostas candidatas gerada pelas etapas anteriores. Sem a seleção da resposta, um sistema de QA não difere em muitos aspectos de um buscador, deixando que o usuário busque a resposta correta entre inúmeras candidatas, não trazendo nenhuma grande inovação ou melhoria. A seleção da resposta pode ser feita com diversas técnicas existentes, a grande maioria delas independente da língua utilizada pelo sistema, mas qual é o impacto da utilização de cada uma destas técnicas no sistema como um todo? Existe alguma diferença significativa do desempenho de cada uma das técnicas em línguas diferentes?

Para responder estas perguntas, este trabalho pretende avaliar diversos métodos de seleção da resposta e comparar seus resultados em um sistema para a língua inglesa e em um sistema equivalente para a língua portuguesa. Pretendemos verificar qual a diferença no desempenho das técnicas em diferentes línguas, e também utilizá-las em um sistema com maior número de recursos para verificar o impacto da utilização desses recursos na etapa de seleção da resposta.

Para isso desevolvemos um sistema de Perguntas e Respostas para o português e um sistema para o inglês, com a mesma arquitetura, e avaliamos cada uma das técnicas de seleção da resposta nos dois sistemas. Para a comparação com o sistema da língua inglesa com mais recursos, utilizamos o OpenEphyra[1] até sua parte de geração das respostas candidatas e aplicamos os algoritmos de seleção da resposta na saída do sistema.

Este trabalho está estruturado da seguinte maneira, no capítulo 2 será apresentado o estado da arte do *Question Answering*, aprofundaremos mais cada uma das etapas e os principais sistemas de QA existentes. No capítulo 3 apresentaremos as tecnologias utilizadas em sistemas de QA existentes e quais delas utilizaremos neste trabalho. O capítulo 4 apresentará os sistemas desenvolvidos para a avaliação dos métodos de seleção da resposta estudados. O capítulo 5 mostrará os resultados obtidos e a análise deles e, por fim, no capítulo 6, concluíremos e exploraremos os trabalhos futuros na área de QA.

[1] OpenEphyra disponível em http://www.ephyra.info/

2 TRABALHOS RELACIONADOS

Neste capítulo é apresentado o estado da arte na área de sistemas de Perguntas e Respostas, que serve como fundamentação teórica e como motivação para a realização desse trabalho. Este capítulo é dividido em quatro seções, na primeira é descrito um sistema de QA e suas etapas, na segunda são apresentados trabalhos descrevendo sistemas de QA em inglês e em português, na terceira apresentaremos trabalhos descrevendo a etapa de maior interesse desse trabalho, a seleção da resposta, e por fim, na última seção, falaremos sobre a avaliação de sistemas de Perguntas e Respostas.

2.1 *Question Answering*

Além dos sistemas de QA de domínio aberto que são o foco desse trabalho existem sistemas especializados nas mais diversas áreas: biologia (SOK-I), informações biográficas (BioGrapher), turismo (WEBCOOP), previsão do tempo, finanças, domínios médicos, domínios geográficos, jogos de interpretação de papéis, entre outros, segundo (MOLLA; VICEDO, 2007). Um sistema de QA pode ser dividido em quatro etapas: análise da pergunta, identificação dos documentos candidatos, geração das respostas candidatas, seleção da respostas, como visto na Figura 2.1.

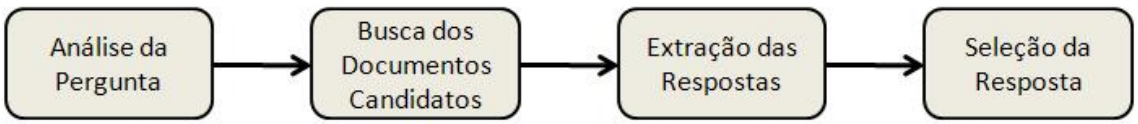

Figura 2.1: *Pipeline* padrão de um sistema de QA

A seguir detalharemos cada uma destas etapas e apresentaremos os principais sistemas de QA.

2.1.1 Análise da Pergunta

A primeira etapa de um sistema de Perguntas e Respostas, a análise da pergunta, tem como entrada a pergunta em linguagem natural feita pelo usuário, essa pergunta é automaticamente analisada por um anotador, normalmente um parser, que identifica características da pergunta que serão utilizadas nas próximas etapas, dentre estas destacamos as palavras chave e o verbo principal da frase. Na figura 2.2 podemos ver um resumo do que acontece nessa etapa e onde ela se encaixa no pipeline de um sistema de QA.

A principal informação obtida na etapa de análise da pergunta é o tipo de resposta esperada. Esta etapa normalmente utiliza métodos superficiais para a língua inglesa, pois

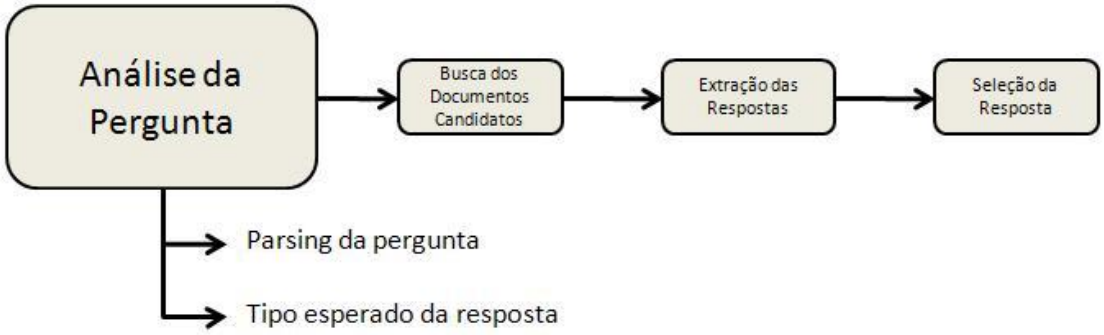

Figura 2.2: A análise da pergunta no pipeline de um QA

somente analisando o pronome interrogativo utilizado na pergunta é possível prever o tipo de resposta esperado. Existem diversas taxonomias de perguntas utilizadas para identificar o tipo de pergunta, porém a mais utilizada é chamada de *wh-words* ou *wh-phrases* devido à utilização das expressões interrogativas da língua inglesa (*who, which, what, when, where, why, how*). Como exemplo de uma taxonomia temos a Tabela 2.1 de (SRIHARI; LI, 2000):

Tabela 2.1: Tipos de resposta esperados

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

Uma abordagem mais elaborada para a análise da pergunta é utilizar algoritmos de aprendizado de máquina para decidir qual será o tipo de resposta esperado. A aplicação desses métodos depende de um corpus de perguntas anotadas com seus tipos de resposta esperados para que seja possível a aplicação dos algoritmos e o aprendizado de um modelo que a partir da análise da frase pelo parser gere o tipo de resposta esperado.

Existem alguns sistemas que utilizam processamentos mais profundos e hierarquias de tipos de perguntas mais complexas resultando em um tipo de resposta mais específico.

Quanto mais específico for o tipo de resposta, mais fácil será de encontrar a resposta correta dentro dos documentos, porém caso este tipo seja inadequado para a pergunta pode ser impossível de encontrar a resposta correta. Há também necessidade maior de dados com perguntas anotadas caso a hierarquia utilizada seja maior e mais específica, pois é necessário um grande número de exemplos de cada tipo de pergunta, e quanto mais tipos nossa hierarquia tiver, mais exemplos são necessários para o algoritmo de aprendizado de máquina.

2.1.2 Busca dos Documentos Candidatos

A busca dos documentos candidatos tem como entrada a pergunta e as características dela, descobertas na etapa anterior, e gera uma lista com os documentos mais prováveis de conter a resposta. Para sistemas baseados na web essa etapa é normalmente feita com uma ferramenta de busca, a mais utilizada é o Google[1] por ter uma maior cobertura e a melhor função de *rank*, segundo (KWOK; ETZIONI; WELD, 2001). A busca pelos documentos candidatos também pode ser feita em uma base documentos locais, que normalmente é feita pelo Lucene[2].

Uma importante tarefa dentro da etapa de busca dos documentos candidatos é a reformulação da consulta. Nela, a pergunta feita pelo usuário é modificada e transformada em uma consulta em forma canônica que será feita à ferramenta de busca. Muitos sistemas de QA tem um módulo de reformulação de consulta, como o MULDER (KWOK; ETZIONI; WELD, 2001), o AnswerBus (ZHENG, 2002) e o Qualim (KAISSER, 2005). Ela pode ser a mesma pergunta submetida pelo usuário, a pergunta sem *stopwords*[3], com o radical de algumas palavras extraído no processo chamado *stemming* (por exemplo, o *stemming* das palavras *fisher, fishing, fished* produz a palavra *fish*). Também consultas mais elaboradas podem ser feitas contendo sinônimos separados pelo operador lógico OR, como *killed OR murdered*. Há ainda outras alternativas de consultas que procuram reorganizar as palavras e submeter uma consulta no formato em que as respostas aparecerão nos textos, como por exemplo no Qualim (KAISSER, 2005), onde a pergunta *"When was Amtrak founded?"* é reformulada para *"Amtrak was founded in ANSWER"* e *"In ANSWER Amtrak was founded"*. Mais sobre o Qualim na seção 2.3.1.5.

Figura 2.3: A busca dos documentos candidatos no *pipeline* de um QA

[1]Google, acessível em http://www.google.com.br

[2]Lucene, disponível em http://lucene.apache.org/

[3]Stopwords são palavras consideradas não relevantes. Palavras muito comuns na língua como artigos e preposições

2.1.3 Geração das Respostas Candidatas

A etapa de Geração das Respostas Candidatas, ou Extração da Resposta, é responsável por analisar os textos retornados pela etapa anterior e procurar por frases aonde a resposta possa estar. Como será visto na Seção 2.3.1.1, geralmente o texto é dividido em frases e como seria um processo extremamente custoso e demorado analisar todo o texto, somente as frases que contém alguma das palavras chave da pergunta são analisadas por um parser e por um reconhecedor de entidades nomeadas, segundo (KWOK; ETZIONI; WELD, 2001). Também podem ser extraídas as frases adjacentes àquelas que contém as palavras chave da pergunta com o objetivo de aumentar as chances de extrair a resposta correta. Neste ponto os sistemas apresentam uma fronteira não muito clara entre as etapas de geração das respostas candidatas e a seleção da resposta: alguns sistemas extraem somente frases, porém o mais comum é que nessa etapa sejam identificadas as entidades do tipo de resposta esperado.

Após a extração das frases e a análise, são identificados os elementos dessas que são do tipo de resposta esperada. Essa identificação pode nos levar a poucos candidatos para a próxima etapa caso o sistema utilize uma taxonomia de respostas mais complexa, pois teremos somente entidades de um tipo bem específico, ou ainda diversos candidatos caso os tipos presentes na taxonomia sejam mais abrangentes.

Figura 2.4: A extração da Resposta no *pipeline* de um QA

2.1.4 Seleção da Resposta

A seleção da resposta é a última etapa do processo de QA, ela tem a tarefa de atribuir um score a cada uma das respostas candidatas e com este ordená-las para que a resposta correta seja a de maior score.

Nessa etapa diversas técnicas podem ser empregadas, desde técnicas superficiais, como o número de palavras em comum entre a pergunta e a frase de resposta, até técnicas mais profundas como algoritmos de aprendizado de máquina que levam em conta aspectos sintáticos das frases de pergunta e resposta. Os algoritmos desta etapa serão descritos detalhadamente na Seção 2.4 que apresenta o estado da arte em seleção da resposta e na Seção 4.2 que apresenta os algoritmos utilizados neste trabalho.

2.2 Linhas de Pesquisa em Question Answering

Segundo, (BURGER et al., 2001) há 12 linhas de pesquisa em Question Answering. São elas:

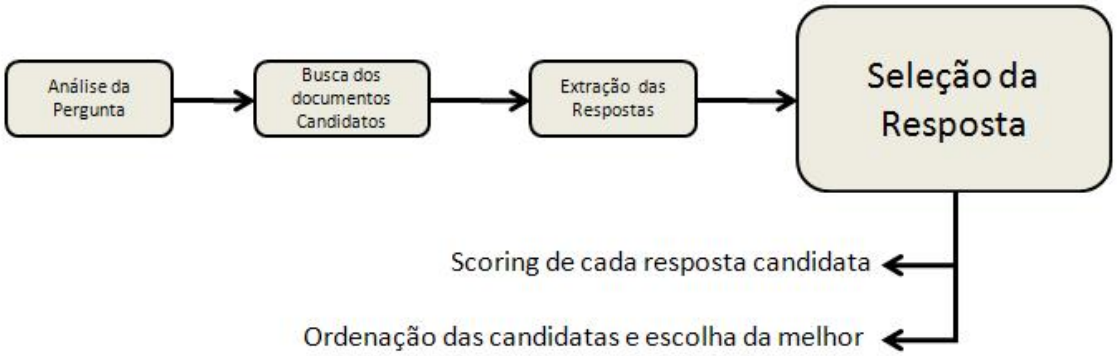

Figura 2.5: A seleção da resposta no *pipeline* de um QA

1. Classes de perguntas: Envolve o estudo da taxonomia das perguntas, da complexidade das perguntas e do estudo do processamento de perguntas através de bases de conhecimento e ontologias.

2. Processamento de perguntas: Estudo de modelos semânticos para o entendimento das perguntas. Estudo de perguntas equivalentes, similares e ambíguas.

3. QA e contexto: As perguntas são feitas dentro de um contexto e esse contexto pode servir para esclarecer a pergunta, resolver ambiguidades e manter a linha de raciocínio em uma série de perguntas.

4. Fontes de dados: Antes de responder a uma pergunta é preciso saber quais fontes de conhecimento estão disponíveis e se eles são suficientes para respondê-la. As fontes podem ser bases de dados, bibliotecas digitais e até mesmo arquivos multimídia.

5. Extração da Resposta: A extração da resposta depende da complexidade da pergunta, do tipo de resposta obtido no processamento da pergunta, da fonte de dados e foco da pergunta e do contexto. Essa área envolve o estudo de métricas de avaliação, de completude, de correção e de justificativas para a resposta.

6. Formulação da Resposta: A resposta deve formulada da maneira mais natural possível. Para isso é preciso estudar como criar uma resposta com informações vindas de diferentes fontes, evitando informações sobrepostas e contraditórias. Essa área também envolve a modelagem de fatos e eventos atuais ou passados, pois dependendo do momento da pergunta, a resposta pode ser diferente.

7. QA em tempo real: É preciso desenvolver sistemas que respondam em poucos segundos independentemente da complexidade da pergunta e otimizar as etapas que são os gargalos de um QA: a busca dos documentos e a extração das respostas.

8. QA multilíngue: Desenvolver um QA que responda perguntas em diversas línguas e busque suas respostas em textos escritos em uma língua diferente da qual a pergunta foi feita. Para isso é necessário o desenvolvimento de *part-of-speech* (PoS)[4] taggers, parsers e reconhecedores de entidades nomeadas em várias línguas. Também é preciso utilizar tradução automática para traduzir as perguntas, e ontologias e bases de conhecimento independentes de língua.

[4]PoS, part-of-speech são as classes gramaticais das palavras

9. QA interativo: O usuário pode desejar ter um diálogo com o sistema, em casos nos quais as informações necessárias não foram identificadas pelo sistema. Nesse caso é preciso desenvolver modelos de diálogo que contenham detecção de intenções, objetivos e planos em comum com o usuário e resolução de coreferências.

10. Raciocínio avançado para QA: Alguns usuários podem desejar respostas que estejam fora do escopo de textos e bases de dados estruturadas. Para que os sistemas sejam capazes de responder esse tipo de questão eles devem poder integrar diversos componentes utilizando raciocínio e senso-comum. É também preciso ter capacidade de inferir fatos novos e montá-los para gerar uma resposta.

11. Perfil de usuário: O perfil de usuário guarda informações a respeito do contexto, do domínio de interesse, esquemas de raciocínio normalmente utilizados, entre outros.

12. QA colaborativo: Estuda o desenvolvimento de modelos que detectam usuários no mesmo contexto e também modelos que buscam perguntas não relacionadas que tem respostas relacionadas e vice-versa.

2.3 Principais Trabalhos

Nesta seção apresentaremos alguns dos principais trabalhos sobre QA. Ela está dividida em duas subseções, primeiro apresentando os trabalhos para QA em inglês e depois os trabalhos para QA em português.

2.3.1 QA inglês

Nesta seção serão descritos o MULDER (KWOK; ETZIONI; WELD, 2001), o AnswerBus (ZHENG, 2002), o Javelin (NYBERG et al., 2002), o sistema da Universidade de Michigan (QI et al., 2002), o Qualim (KAISSER, 2005) e o OpenEphyra.

2.3.1.1 Estendendo sistemas de Perguntas e Respostas para a Web

QAs baseados na web apresentam os seguintes desafios:

1. Formar as consultas corretas: uma consulta muito geral pode recuperar muitas páginas e uma consulta muito restrita pode não retornar nenhuma página.

2. Ruído: mesmo com um conjunto correto de palavra chave submetidas à ferramenta, podem ser recuperadas páginas que falem sobre outros assuntos.

3. Limitação de recursos: Apesar da velocidade das ferramentas de busca, é muito custoso utilizar muitas consultas para responder a uma pergunta. Por isso um sistema de QA deve submeter as consultas certas, pois deve considerar o tempo que o usuário está disposto a esperar pela resposta.

Neste sentido, o MULDER (KWOK; ETZIONI; WELD, 2001) é um sistema que foca em perguntas factóides e busca respostas na web. Perguntas factóides são aquelas que podem ser respondidas com um fato, isto é um nome, um local ou uma data, como *"Quem é o pato mais rico do mundo?"*, *"Onde nasceu Carl Barks?"* ou *"Quando nasceu Thomas Mann?"*. Embora elas sejam uma parcela limitada das consultas feitas às ferramentas de busca, é uma oportunidade de reduzir o esforço do usuário para encontrar a resposta. Como se trata de um sistema em tempo real, um fator importante para o desenvolvimento

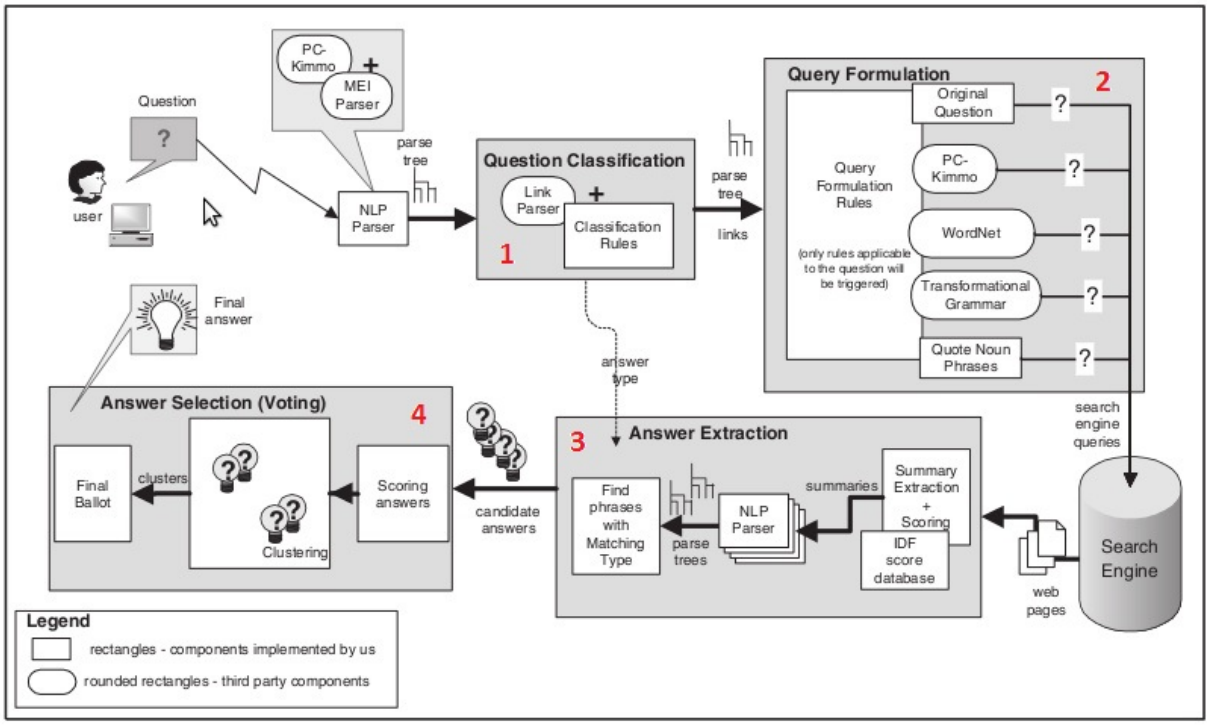

Figura 2.6: Arquitetura do MULDER

deste sistema é o tempo de resposta. O MULDER submete várias consultas a uma ferramenta de busca (Google) e extrai respostas dos resultados obtidos. A arquitetura do MULDER é apresentada na figura 2.6. Nela, a pergunta submetida pelo usuário é analisada por um parser [5] de linguagem natural, que constrói uma árvore com a estrutura sintática da pergunta. O classificador da pergunta (1) usa esta árvore para determinar o tipo de resposta esperado. Após, o formulador de consulta (2) utiliza a árvore para traduzir a pergunta em uma série consultas que são submetidas em paralelo à ferramenta de busca. O módulo de extração da resposta (3) extrai trechos relevantes e gera uma lista de respostas candidatas. Por fim, o seletor de respostas (4) ordena as respostas.

A classificação da pergunta serve para diminuir o número de respostas candidatas, ele reconhece três tipos de perguntas (nominais, temporais e numéricas), relacionadas a quatro tipos de *wh-phrases*:

1. *Wh-adjetives*, como *"How many millions of Jews were killed in the Second World War?"* e *"How high is the K2?"* esperam uma resposta numérica.

2. *Wh-adverb*

   When, como *"When was Yasser Arafat awarded the Nobel Peace Prize?"*, espera uma resposta temporal.

   Where, como *"Where was Nelson Mandela born?"* espera uma resposta nominal.

[5] O *parsing* da pergunta é feito com o MEI (*Maximum Entropy-Inspired*) parser que emprega técnicas estatísticas integrado ao analizador léxico PC-KIMMO, que juntos conseguem analizar corretamente todas as questões do TREC-8 (VOORHEES, 1999)

3. *Wh-noun*, geralmente começam com *what*, o MULDER utiliza o Link parser (GRINBERG; LAFFERTY; SLEATOR, 1995) para obter o objeto da verbo na frase e busca na hierarquia de hiperônimos da WordNet [6] (MILLER et al., 1990) pelo tipo desse objeto. Se o objeto for do tipo medida, classificada a pergunta como numérica, se for do tipo tempo, classificada a pergunta como temporal ou então classifica como nominal.

4. O último tipo é caso a pergunta não contenha uma *wh-phrase*, como *"Name a Japanese vulcano."* é classificada como nominal.

A formulação da consulta busca converter a pergunta em um conjunto de palavras chave para uso com o buscador. Há uma boa chance de sucesso se for possível determinar como a resposta aparecerá nos resultados, para isso são empregadas diversas técnicas que transformam a pergunta em diferentes consultas. Quanto mais palavras e restrições na consulta, menos páginas serão encontradas, caso nenhuma página seja encontrada uma consulta menor e mais simples é enviada, por exemplo, sem aspas em uma *noun phrase* (NP)[7], como "question answering".

O módulo de extração da resposta é responsável por adquirir respostas candidatas das páginas da web. Primeiro são extraídos trechos das páginas próximas das regiões contendo as palavras chave (são extraídas algumas frases, pois consumiria muito tempo analisar todas as frases de todas as páginas da web e os resultados de se analisar apenas uma frase são ruins). Os trechos são ordenados de acordo com a importância das palavras (é utilizado a frequência inversa do documento, *inverse document frequency*, detalhada na Seção 4.2.5) contidas nele e sua proximidade. Os n melhores trechos são analisados e são extraídas as respostas candidatas do tipo de resposta esperado.

O seletor de resposta agrupa as respostas candidatas em *clusters* (usando uma versão simplificada do *suffix tree clustering* (ZAMIR; ETZIONI, 1999)) para reduzir a possibilidade de encontrar informações falsas e juntar diferentes formas de escrita da resposta. O score de cada resposta candidata é calculado através da soma dos scores de cada *cluster* e, por fim, são mostrados os resultados com a resposta com maior score dentro do seu cluster.

2.3.1.2 AnswerBus

O AnswerBus (ZHENG, 2002) é um sistema que responde a questões feitas em 6 línguas diferentes e busca suas respostas em 5 ferramentas (Google, Yahoo[8], WiseNut[9], AltaVista[10] e Yahoo News[11]). Primeiro um módulo verifica se a pergunta está em inglês, caso não esteja, traduz a pergunta utilizando o BabelFish[12]. O resto do processo é composto por 4 passos: (1) selecionar ferramentas de busca mais adequadas à pergunta e formar a consulta; (2) obter os documentos; (3) extrair as sentenças; (4) ordenar as respostas e retorna as melhores para o usuário.

Para determinar qual das ferramentas de busca será utilizada, o AnswerBus criou um modelo para determinar quais as mais adequadas para cada consulta. Para isso foram

[6] A WordNet é uma base léxica das palavras da língua inglesa. Ela provê informações sobre similaridade sintático semânticas e será mais detalhada na Seção 2.6.1.1

[7] NP, noun phrase, em português sintagma nominal

[8] Yahoo, acessível em www.yahoo.com

[9] WiseNut, acessível em http://en.wisenut.com/

[10] AltaVista, acessível em http://www.altavista.com/

[11] Yahoo News, acessível em http://news.yahoo.com/

[12] BabelFish, disponível em http://www.babelfish.com

feitas consultas a todas ferramentas para 2000 questões de exemplo e montada uma tabela com o número de possíveis respostas retornadas por cada ferramenta para cada palavra nas perguntas. Ele usa as ferramentas que retornam mais respostas para determinadas palavras na pergunta. Caso as palavras da pergunta não estejam na tabela, é utilizada a média de todas as palavras indexadas.

Uma consulta deve ser bem formulada para obter a resposta correta das ferramentas de busca e também diminuir o tempo de processamento para encontrar esta resposta. A abordagem utilizada consiste de expandir a consulta utilizando sinônimos, o que aumenta a chance de conseguir a resposta correta mas também aumenta o tempo. O AnswerBus foca em encontrar uma consulta que seja boa o bastante (não necessariamente a melhor) e muito rápida. Várias abordagens são utilizadas para formar a consulta:

1. Remover *stopwords* - preposições, pronomes, determinantes, conjunções e interjeições.

2. Uso de tabela de frequência de palavras - Para consultas longas, são removidas palavras muito usadas na linguagem.

3. Modificação de forma - Algumas palavras têm sua forma modificada, principalmente verbos, por exemplo, *end* é convertido para *ended* e *have* para *has*.

Após enviar a consulta e obter os documentos, todas as frases são processadas e filtradas de acordo com o número de palavras em comum com a consulta. As frases restantes são consideradas candidatas a resposta e cada uma recebe um score na filtragem, porém ele é muito fraco para dar boas respostas, assim outros aspectos são levados em conta para determinar a resposta final. É levado em conta o tipo da pergunta (*wh-questions* e *how-questions*), além disso, é utilizado um dicionário específico para QA que contém informações sobre a relação de palavras entre perguntas e respostas. Também são extraídas as entidades nomeadas somente do tipo da pergunta. É utilizada resolução de correferência entre frases adjacentes. É levada em conta a posição do documento retornado pela ferramenta de busca e cada ferramenta produze um score diferente para seus documentos. O AnswerBus foi avaliado com as perguntas do TREC-8 e obteve 60% de acertos.

2.3.1.3 University of Michigan at TREC2002

O sistema NSIR da universidade de Michigan (QI et al., 2002) possui três etapas: classificação da pergunta, extração das respostas e ordenação das respostas. Para a classificação do tipo de pergunta foi desenvolvida uma taxonomia hierárquica com 141 tipos, que ajuda a localizar as respostas dentro dos textos. Devido a dificuldade de classificar uma pergunta com somente um tipo, então é utilizado um classificador de questões probabilístico que atribui pesos normalizados a cada tipo possível de pergunta. Essas probabilidades são levadas em conta na ordenação das respostas. É obtida uma lista de frases dos documentos mais relevantes e são computadas as seguintes características de cada frase: (1) frequência, (2) sobreposição, (3) comprimento, (4) proximidade, (5) PoS, (6) assinatura léxica, (7) lista de palavras, (8) entidades nomeadas, (9) *ranking web*.

A ordenação das respostas é feita com um score que é uma combinação linear das características acima listadas e cada tipo de pergunta tem pesos diferentes para cada característica. A resposta com o melhor score é a resposta final para a pergunta.

No TREC2002 existem perguntas que devem ter a resposta marcada como nula, pois não há resposta no corpus. Para encontrar essas respostas foi definido um limiar para

o score e respostas com score final abaixo desse limiar são respondidas como nulas. Esse score foi aprendido com base em questões do TREC-10. Os resultados obtidos no TREC2002 não foram bons com menos de 20% de acertos.

2.3.1.4 Javelin

O Javelin (NYBERG et al., 2002) é um sistema de QA de domínio aberto. Ele possui 4 etapas de processamento:

1) Análise da Pergunta: A partir da pergunta de entrada gera um objeto contendo a clasificação do tipo da pergunta e da resposta de acordo com uma taxonomia pré-definida, uma lista de palavras chave e suas formas alternativas.

2) Recuperação de Documentos: Primeiro é utilizado um reconhecedor de entidades nomeadas nos documentos, depois a busca é feita usando as palavras chaves e o tipo de resposta esperado obtidos na análise da pergunta. Essa busca é feita em um processo gradativo: primeiro uma consulta muito restrita é feita e caso não retorne o número de documentos esperado algum parâmetro é modificado para que mais documentos sejam retornados. A saída é uma lista ordenada de documentos.

3) Extração das Respostas Candidatas: O objetivo desse módulo é identificar passagens do texto onde se encontram as respostas candidatas e atribuir um score ao par passagem-resposta. Um classificador é utilizado para atribuir esses scores. Ele leva em conta diversos atributos da passagem-resposta como análise de PoS e identificação de entidades nomeadas.

4) Seletor de Resposta: Este módulo produz uma lista de respostas ordenadas por um score de confiança e devolve como resposta final aquela com score mais alto. Para isso, as respostas são colocadas em uma forma canônica (dependente do tipo de resposta, há uma forma canônica para datas, uma para nomes próprios, uma para localidades). Depois, o score recebido do extrator de respostas é normalizado entre 0 e 1 usando uma distribuição normal. As respostas são agrupadas em *clusters*. Da mesma forma que a canonização, há uma forma de agrupar cada tipo de resposta. Por fim é calculado um score de confiança para cada *cluster* de respostas e a resposta mais específica do *cluster* de maior score é escolhida como a correta.

2.3.1.5 Qualim

O Qualim (KAISSER, 2005) é um sistema baseado em refraseamento. Ele utiliza padrões para reformular as perguntas para respostas em potencial. Por exemplo, a pergunta *"When was Amtrak founded?"* é reformulada para *"Amtrak was founded in ANSWER"* e *"In ANSWER Amtrak was founded"*. É feita uma busca no Google e a resposta é extraída. Caso nenhum padrão seja identificado é feita uma busca por palavras chave da pergunta: os *n-grams*[13] dos resultados são analisados e o mais frequente é retornado.

Inicialmente os padrões de resposta foram criados manualmente, mas isto é um processo muito trabalhoso, portanto a idéia foi utilizar uma base de recursos léxicos como a FrameNet (BAKER; FILLMORE; LOWE, 1998) para criar padrões automaticamente. A FrameNet tem por objetivo de criar uma amostra de como a linguagem natural funciona. Ela é uma base de dados léxica baseada em padrões semânticos e suportada por evidências em corpus. A FrameNet é usada no Qualim para criar um conjunto de consultas exatas para uma ferramenta de busca, analisar as frases retornadas pela ferramenta de busca e encontrar as respostas exatas para a pergunta.

[13] N-gram é uma sequência contínua de n items em um texto

No Qualim, primeiro uma pergunta é analisada pelo parser MiniPar (LIN, 1998) e são descobertas informações como o verbo principal da frase, o sujeito e os objetos. Com essas informações é possível pesquisar na FrameNet por frases semelhantes. Assim é possível identificar quais papeis semânticos são designados a quais papéis sintáticos nas frases.

Por exemplo, a pergunta *"When was the telegraph invented?"* tem como verbo principal *invented*, que possui duas entradas na FrameNet: uma delas é *"Du Pont had invented nylon in the late 1930s..."* que tem anotados os papéis semânticos de *Du Pont* como *Cognizer* e *nylon* como *Invention*, analisando essa frase com o MiniPar é identificado que usualmente o *Cognizer* é um NP na posição de sujeito e *Invention* é um NP na posição de objeto. Com essas informações é possível saber que em uma frase que potencialmente contenha a resposta, ela será o sujeito, o verbo será *invent* e o objeto da frase deverá ser *the telegraph*. Assim é montado um padrão em que uma pergunta da forma: *"Who invented OBJECT?"* é respondida por *"ANSWER had invented OBJECT?"*.

Esse algoritmo de criação de padrões através da FrameNet foi utilizado no TREC 2005 em uma versão inicial e conseguiu responder 35 das 362 perguntas sobre fatos, 25 corretamente.

2.3.1.6 OpenEphyra

O OpenEphyra ou Ephyra[14] é o primeiro framework aberto para QA, sua arquitetura pode ser vista na Figura 2.7.

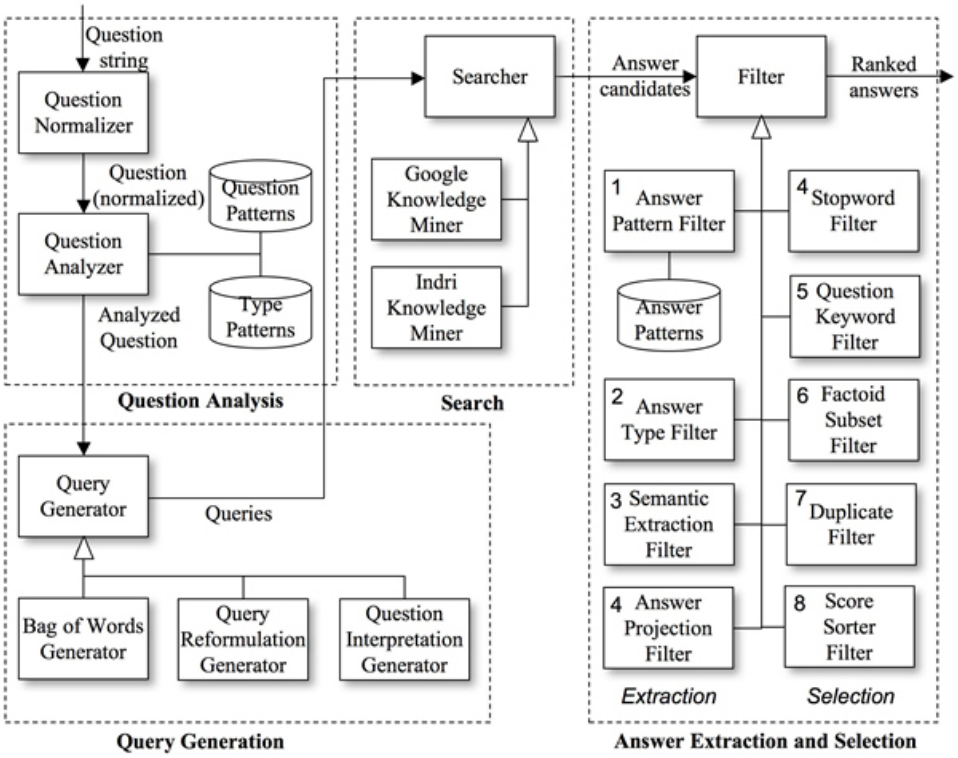

Figura 2.7: Arquitetura do OpenEphyra

O Ephyra possui um módulo de análise da pergunta que utiliza uma hierarquia de 154 tipos de resposta esperado, com 44 categorias de nível mais alto para classificar a

[14]OpenEphyra disponível em http://www.ephyra.info/

pergunta. O Módulo de Geração de consulta utiliza a WordNet para expandir a consulta, que é feita à web para recuperar os documentos relevantes. O Ephyra apresenta um aspecto um pouco diferente dos demais sistemas em seu pipeline: as etapas de extração e seleção da resposta são combinadas e isto se deve ao fato que elas são operações semelhantes. No Ephyra estas etapas são implementadas por meio da aplicação de diversos "filtros", atuando cada um na saída dos anteriores. Cada filtro tem uma função específica, o primeiro, por exemplo, que atua sobre as frases completas extraídas dos documentos, procura por respostas do tipo esperado. Alguns filtros tem funções simples, como eliminar respostas contidas na pergunta, por exemplo eliminar a resposta candidata *JFK* para a pergunta *"Who killed JFK?"*, enquanto outros tem a tarefa de atribuir scores de relevância às respostas, fazendo busca na web em geral e na Wikipedia. Também são utilizados Gazetters[15] e a WordNet para atribuir scores às respostas.

2.3.2 QA português

Nesta seção serão descritos alguns sistemas de QA para o português. Serão apresentados o Priberam (AMARAL et al., 2005), o sistema de QA para o português com o melhor desempenho no CLEF (ver seção 2.5.1) e o Comunica (WILKENS et al., 2010). Outros sistemas de QA para o português são o IdSay (CARVALHO; MATOS; ROCIO, 2010) e a o desenvolvido na Universidade de Évora (SAIAS; QUARESMA, 2007).

2.3.2.1 Priberam

O sistema de Perguntas e Respostas de Priberam é basedado no módulo de português do TRUST - *Text Retrieval Using Semantic Technologies* - um projeto que visa desenvolver uma ferramenta de busca semântica multilíngue capaz de responder perguntas em linguagem natural em inglês, francês, italiano, polonês e português. No TRUST, o sistema procura uma resposta em uma coleção de documentos e retorna uma lista ordenada de frases que contêm a resposta. A única diferença para o QA@CLEF[16] é que deve ser extraída uma única resposta exata. Para o desenvolvimento do FLiP[17], uma série de recursos como dicionários, corretores ortográficos e outras ferramentas de linguagem, desenvolvidos por Priberam e do módulo de português do TRUST foram necessários uma base de recursos léxicos, ferramentas de softwares, informações estatísticas extraídas de corpus e outros recursos adaptados para a tarefa de Question Answering.

Os recursos léxicos utilizados incluem um dicionário de ampla cobertura, um *thesaurus*, que provê um conjunto de sinônimos para cada unidade léxica, e uma ontologia multilíngue. As ferramentas de software incluem o Priberam SintaGest (AMARAL et al., 2004), uma ferramenta interativa que permite a construção e o teste de gramáticas para qualquer língua, que foi utilizado para a construção de gramáticas de português europeu e brasileiro.

O sistema utilizado no CLEF possui 5 etapas: (1) indexação, (2) análise da pergunta, (3) recuperação de documentos, (4) recuperação de frases e (5) extração da resposta.

A indexação é um processo off-line, de análise dos documentos para coleta de informações para um arquivo de indexação, que facilitará a recuperação dos documentos.

A análise da pergunta recebe como entrada uma pergunta que é analisada, depois é feita a categorização. Para a categorização da pergunta foram desenvolvidas uma série

[15] *Gazetters* provem informações geográficas que permitem a identificação de países, suas cidades, estados, capitais e etc.

[16] QA@CLEF, trilha de *Question Answering* do CLEF, ver Seção 2.5.1

[17] FLiP, Ferramentas para a língua Portuguesa, descritos em www.flip.pt

de regras contextuais que classificam a pergunta em uma ou mais de 86 categorias. Também são associados a pergunta padrões de pergunta-resposta (*Question Answer Patterns*, QAP's), que identificam onde uma resposta ocorre dentro de uma frase. Cada QAP em que a pergunta foi classificada recebe um score de adequação. O próximo passo é a extração de pivôs: os elementos chave da pergunta, que podem ser palavras, expressões, entidades nomeadas, números, datas, etc. Para cada pivô são armazenadas diversas informações como PoS e sinônimos(obtidos no *thesaurus*). Também são obtidos os domínios ontológicos e terminológicos da pergunta. Todas essas informações alimentam o módulo de recuperação dos documentos.

A recuperação dos documentos calcula um score para cada documento baseados nas informações da pergunta e obtem os 30 com maior score.

A recuperação das frases analisa as frases dos documentos cujos pivôs correspondem aos da pergunta. É calculado um score levando em conta o número de pivôs correspondentes, o número de sinônimos correspondentes e a proximidade dos pivôs, entre outras características. As frases com score menor que um certo limiar são descartadas e as restantes são enviadas ao próximo módulo.

O módulo de extração da resposta aplica cada QAP nas frases recebidas e extrai as respostas de cada frase. A cada resposta é calculado um score levando todas as características anteriormente computadas e a adequação ao QAP. O último passo é aumentar os scores das palavras repetidas nas respostas e escolher a resposta com score mais alto como correta.

O sistema de QA de Priberam obteve 64.5% de acurácia no CLEF em 2004, e foi o melhor sistema de QA para o português. Porém ele utiliza muitos recursos e ferramentas para análise profunda que não estão disponíveis em geral, tais como ontologias, *thesaurus* e dicionários.

2.3.2.2 Comunica

O projeto Comunica (WILKENS et al., 2010) é um sistema de Perguntas e Respostas de domínio fechado para o português, que objetiva responder perguntas sobre transferências constitucionais de municípios via telefone. Nele, tanto a pergunta do usuário quanto a resposta do sistema são em liguagem natural, visando a uma maior inclusão digital.

A arquitetura se divide em quatro módulos: reconhecimento de voz, processamento da linguagem natural, acesso a banco de dados e síntese de voz.

O módulo de reconhecimento de voz realiza a conversão de áudio para texto. O processamento da linguagem natural tem a função de identificar dados relevantes informados pelo usuário a partir da frase transcrita (pelo módulo de reconhecimento de voz). A identificação dos conceitos é feita por meio de duas ontologias que validam as palavras da pergunta do usuário: uma de propósito geral e uma do domínio da aplicação. Os conceitos identificados são então buscados pelo módulo de acesso a banco de dados e a resposta gerada é sintetizada pelo módulo de síntese de voz.

A arquitetura do Comunica difere da arquitetura mais comum em sistemas de QA pois o Comunica faz a busca em uma base estruturada de dados ao contrário dos sistemas apresentados anteriormente que fazem busca em textos. Comparando com a arquitetura comum de um sistema de QA, o módulo de processamento da linguagem faz a análise da pergunta e o módulo de acesso a banco de dados faz a extração e seleção da resposta utilizando as informações obtidas na análise da pergunta. A etapa de busca não existe pois há somente um banco de dados de onde as informações são extraídas.

2.4 Seleção da Resposta

Nesta seção são apresentados alguns trabalhos sobre a etapa de seleção da resposta em um QA. Primeiro abordamos métodos probabilísticos de seleção da resposta, depois métodos lógicos e por fim métodos de aprendizado de máquina.

2.4.1 *Framework* probabilístico para seleção da resposta

A seleção da resposta é uma etapa muito desafiadora de um sistema de QA, que geralmente envolve identificar a resposta correta entre inúmeras opções. Para selecionar a resposta correta de uma pergunta precisamos avaliar dois aspectos das respostas candidatas:

1. Validação das Respostas: como identificar uma resposta correta entre as incorretas.

2. Similaridade das Respostas: como explorar as evidências de similaridade entre as respostas, como quando há respostas redundantes ou representando a mesma entidade na lista de respostas candidatas.

No trabalho de (KO; SI; NYBERG, 2007) é proposto um *framework* probabilístico que estima a probabilidade de uma resposta estar correta usando diversas características de validação das respostas e similaridade entre as respostas candidatas. As características de validação das respostas utilizadas são:

1. Características baseadas em conhecimento: são utilizados Gazetters e a WordNet.

   A partir da informação dos *Gazetters* é gerado um score entre -1 e 1 para uma resposta candidata, 0 significa que o *Gazetter* não contribuiu para a validação da resposta. Um score de -1 significa que a resposta não é do tipo semântico esperado (por exemplo, uma resposta candidata com o nome de um país para uma pergunta "Qual cidade..."), e 1 significa que o tipo é o esperado.

   A partir da WordNet também é atribuído um score entre -1 e 1 para um resposta candidata, calculado de forma semelhante aos *Gazetters*.

2. Características baseadas em dados: São utilizados o Google e a Wikipedia.

   Wikipedia: É utilizado um algoritmo que calcula o tf-idf[18] da página da Wikipedia da resposta candidata. Este algoritmo foi utilizado neste volume e está detalhado na seção 4.2.6

   Google: Uma consulta consistindo de uma resposta candidata e as palavras chave da pergunta são submetidas ao Google para gerar um score numérico. Mais detalhes na seção 4.2.4

As características baseadas em similaridade das respostas utilizadas foram divididas em métricas de distâncias de texto e lista de sinônimos. Esse *framework* foi utilizado em conjunto com o Javelin (NYBERG et al., 2002) e entre as características de validação utilizadas as que obtiveram melhores resultados foram as orientadas a dados (Wikipedia e Google). As características de similaridade com melhor desempenho foram as listas de sinônimos. Nos testes realizados, as combinações de todas as características de validação e das de similaridade apresentaram uma melhora nos resultados do sistema.

[18] Tf-idf, *term frequency* e *inverse document frequency*, ver Seção 4.2.5

2.4.2 Validação lógica

O MAVE (GLOCKNER; HARTRUMPF; LEVELING, 2007) é um sistema que tem como objetivo filtrar e combinar resultados de diferentes sistemas de QA baseado em validação lógica de respostas. O sistema recebe um conjunto de respostas - que incluem a resposta, um trecho do texto que justifica a resposta e um score de confiança - e reordena-as.

Primeiro é feita a análise linguística utilizando o WOCADI parser (HARTRUMPF, 2003) que resulta em uma representação MultiNet (HELBIG, 2005) da pergunta e da resposta. MultiNet é um paradigma de representação de conhecimento e uma linguagem para representação do significado de expressões em linguagem natural baseada em redes semânticas. A representação semântica da MultiNet é traduzida para uma lista de literais que representam a hipótese expressada pela resposta. Essa lista de literais é provada logicamente. Um aspecto importante é que caso a prova não seja possível, literais vão sendo excluídos da lista até que a prova seja possível. Após a prova é computado o nível de erro de cada resposta baseado em diversos fatores, como a qualidade da análise linguística e a quantidade de literais excluídos da prova lógica. O nível de erro é transformado em probabilidade de corretude, que é feito utilizando um modelo probabilístico de erro derivado de um corpus de treinamento.

A probabilidade da resposta estar correta é o score de justificação dela. Após obtê-lo para todas respostas, é obtido um score de justificação agregado para todas respostas equivalentes. O critério final para o reordenamento das respostas e seus trechos de texto justificando-as inclue o score de justificação agregado e outros scores e testes cujo objetivo é evitar respostas muito longas e incompletas e preferir respostas com uma melhor análise linguística e eliminar falsos positivos.

O MAVE foi testado utilizando questões do CLEF e três sistemas de QA com características diferentes. O primeiro era orientado a precisão, o segundo orientado a *recall* e o último especializado em uma classe de questões. A combinação desses sistemas obteve bons resultados quando foi utilizado um modelo de erro para cada um dos sistemas, atingindo 81% de *recall* em comparação ao resultado ótimo.

2.4.3 Sistema de Perguntas e Respostas orientado a estratégia

O trabalho de (OH; MYAENG; JANG, 2012) busca examinar os efeitos de verificação da resposta e o método de incremento de confiança que são o núcleo do *framework* para QAs proposto. Nele, a chamada sequencial automática de múltiplos módulos de QA é determinada por uma estratégia aprendida para cada tipo de pergunta a ser tratada.

O *framework* funciona da seguinte maneira: o módulo de análise da pergunta processa-a obtendo o formato da resposta, o tema da resposta, o alvo da pergunta e a fonte esperada da resposta. Esses dados são passados ao módulo de seleção de estratégia e execução, que, baseado neles chama um módulo de QA adequado ao tipo de pergunta. A resposta retornada pelo primeiro QA é aceita se o seu valor de confiança seja maior que um limiar, caso contrário um segundo módulo de QA é chamado e os valores de confiança das respostas candidatas podem ser incrementados na junção das respostas dos módulos de QA. Esse processo é repetido até que valor de confiança da resposta mais bem rankeada exceda o limiar, ou não existam mais módulos de QA a serem chamados.

Foram usados 260 pares (pergunta, resposta) para treinamento, que fazem parte dos 760 pares usados para avaliação. Para mostrar a eficácia do QA orientado a estratégia, foram testados 5 casos: (1) QA tradicional usando indexação e recuperação de sentenças,

(2) somente o módulo de QA mais apropriado para a pergunta, (3) agrupar as respostas de todos os módulos de QA, (4) utilizar uma estratégia contruída manualmente e (5) QA com a estratégia aprendida automaticamente.

O QA orientado a estratégia (5) teve a melhor performance entre os cinco testes, com melhor eficácia (mais respostas corretas) e eficiência (respostas mais rápidas). Foi constatado que o método orientado a estratégia tem menos chamadas que o método (3) que utiliza todos os módulos de QA disponíveis, já que as respostas dadas pelo primeiro QA, especializado naquele tipo de questão, já ultrapassam o limiar. Também há mais respostas corretas porque as respostas erradas, dadas pelos módulos não especializados são suprimidas. Também foi analisado o efeito de se adicionar mais módulos de QA. No caso (3) o número de chamadas cresce como seria esperado, mas no caso (5) há uma redução no número de chamadas pois os módulos especializados são chamados antes e dão respostas com um alto nível de confiança.

2.4.4 Seleção da Resposta utilizando SVM

O objetivo do trabalho de (SUZUKI; SASAKI; MAEDA, 2002) é mostrar o método de seleção da resposta que usa SVM (*support vector machines*) (CORTES; VAPNIK, 1995). SVM é um método de aprendizado de máquina em que cada entrada é vista como um vetor n-dimensional e tem uma classe -1 ou +1. SVM é baseado em encontrar um hiperplano que separe os vetores da duas classes maximizando a margem entre os exemplos de treinamento negativos (-1) e positivos (+1).

Do ponto de vista do aprendizado de máquina, a seleção da resposta é a tarefa de treinar e classificar candidatos a resposta em corretas e incorretas. O conjunto de exemplos para treinamento consiste de vetores de características das respostas candidatas.

Os três primeiros passos de um QA são feitos normalmente, com a análise da pergunta, a recuperação dos documentos e a extração das respostas candidatas. Após esses passos são extraídas as características das respostas candidatas para a seleção da resposta.

As características utilizadas para treinamento SVM são extraídas usando uma função de janela de tamanho variável [19] a nível de palavra, frase e parágrafo. Essas características são: percentual de correspondência de palavras chave, *part-of-speech*, radicais, inflexões, entidades nomeadas correspondentes com o tipo da resposta entre outras.

A avaliação do método foi feita com validação cruzada com dez conjuntos, nove para treinamento e um para teste. Foram usadas 1358 perguntas do TREC que tinham respostas exatas e o método de SVM apresentou uma performance estatisticamente melhor que outros métodos de aprendizado de máquina como árvores de decisão e máxima entropia.

2.5 Avaliação

Nesta seção são descritas as principais conferências de recuperação de informação que tem uma trilha de QA, e também um artigo que propõe uma técnica para automatizar a avaliação de um sistema de QA.

[19] Uma função de janela de tamanho n analisa os n elemento em torno do elemento atual, por exemplo, uma janela de tamanho 2 a nível de palavra analisará as duas palavras anteriores e as duas posteriores à atual.

2.5.1 TREC e CLEF

O TREC[20] (DANG; KELLY; LIN, 2007) e o CLEF[21](PETERS, 2010) são as duas principais conferências na área de *Question Answering*. O TREC (*Text REtrieval Conference*) tem o propósito de apoiar a pesquisa em recuperação de informação provendo infraestrutura para avaliação em larga escala de metodologias. O TREC existe desde 1992 e tem como língua alvo o inglês, porém foi a primeira conferência a realizar avaliações em larga escala de outras línguas, como o espanhol e o chinês. Os sistemas de QA tiveram um grande avanço desde o início da trilha de QA introduzida no TREC-8 (1999), que foi a primeira avaliação em larga escala de sistemas de QA. No TREC-8 200 perguntas factóides deveriam ser respondidas pelos sistemas com sequências de caracteres longas (de 250 bytes) e curtas (de 50 bytes). A partir do TREC-11 os sistemas passaram a ter que responder com a frase exata de resposta. Em 2003 o TREC começou a utilizar perguntas não-factóides, como definições e listas.

O CLEF (*Cross Language Evaluation Forum*) é outra grande conferência sobre recuperação de informação e conta com uma trilha específica de QA (QA@CLEF), que se subdivide em trilhas específicas para cada língua, incluíndo o português, e também trilhas multilíngue. O CLEF acontece desde 2000 e a trilha de português acontece desde 2004.

2.5.2 Avaliando sistemas de Perguntas e Respostas usando inserção de respostas de FAQs

A avaliação de um sistema de Perguntas e Respostas é um processo muito trabalhoso, que consome muito tempo e exige um especialista. O trabalho de (LEIDNER; CALLISON-BURCH, 2003) propõe uma maneira para automatizar a avaliação dos sistemas de QA que não requer intervenção humana pois utiliza FAQs. Um FAQ (*frequently asked questions*, perguntas frequentemente feitas) é formalmente um conjunto de pares pergunta, resposta. Nesse método, uma resposta é inserida em um documento randômico em uma posição conhecida, depois o sistema é utilizado com as perguntas do FAQ e a avaliação consiste de verficar se as respostas foram recuperadas. Esse método também permite a contínua avaliação de um sistema durante seu ciclo de desenvolvimento. Ele permite a avaliação de cada componente verificando se as respostas conhecidas continuam sendo consideradas em cada componente do sistema. Assim é possível identificar componentes que incorretamente excluem respostas corretas. A figura 2.8 ilustra o processo de inserção de uma resposta em um sistema e a identificação de um componente defeituoso do sistema pelo qual a resposta inserida não passa.

Os problemas conhecidos desse método são:

1. Caso alguma outra resposta também correta esteja presente em algum documento, ela será marcada como errada.

2. A reposta inserida não é relacionada ao assunto do documento em que ela foi inserida e isso pode ser um problema na fase de busca dos documentos.

3. FAQs podem conter muita anáfora e material dependente do contexto, e essas questões podem não ser o tipo de pergunta normalmente feita a sistemas de Perguntas e Respostas.

A grande vantagem desse método é sua natureza totalmente automática para avaliação de um sistema de Perguntas e Respostas.

[20] Site oficial do TREC acessível em http://trec.nist.gov/

[21] Site oficial do CLEF acessível em http://clef-campaign.org/

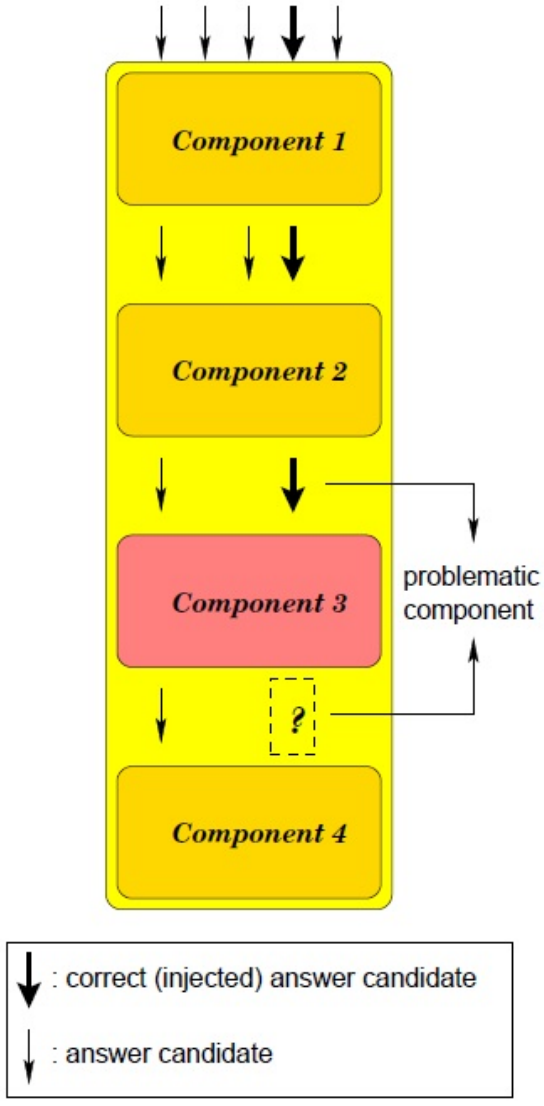

Figura 2.8: Identificação de um componente com erro usando inserção de respostas de FAQ

2.6 Recursos

Um sistema de QA é muito complexo e utiliza diversos recursos externos para obter bons resultados. Nesta seção abordaremos alguns dos recursos utilizados pelos sistemas de QA, como ontologias, parsers e reconhecedores de entidades nomeadas.

2.6.1 Ontologias

Ontologias (GRUBER, 1993), especialmente as de ampla cobertura, são recursos de grande valor, principalmente nas áreas de Inteligência Artificial e Processamento de Linguagem Natural. Uma ontologia é um modelo de dados que descreve um conjunto de conceitos dentro de um domínio e os relacionamentos entre esses conceitos. Em um QA de domínio específico, uma ontologia é o recurso principal do sistema. Porém ontologias também são importantes em QA de domínio aberto. Um exemplo de ontologia de domínio pode ser vista em 2.9, a ontologia de domínio utilizada no Comunica (WILKENS et al., 2010).

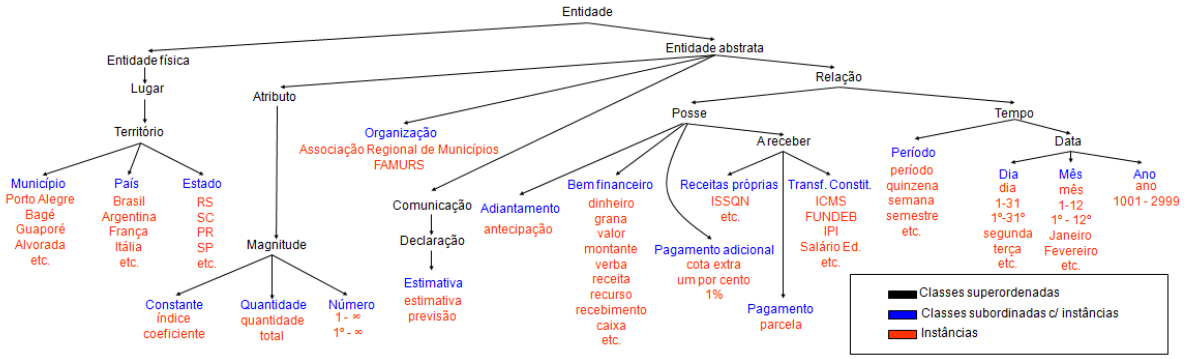

Figura 2.9: Ontologia de domínio do Comunica

2.6.1.1 WordNet

Uma ontologia largamente utilizada em sistemas de domínio aberto para o inglês, é a WordNet (MILLER et al., 1990), uma base de dados léxicos que contém as palavras organizadas em synsets (conjuntos de sinônimos) e contém diversas relações entre esses synsets, como: hiperonímia, hiponímia, meronímia e holonímia.

1. Hiperonímia: Y é um hiperônimo de X se todo X é um (tipo de) Y, felino é um hiperônimo de gato, porque todo gato é um membro da classe dos felinos.

2. Hiponímia é o inverso de hiperoníma, veleiro é um hipônimo de embarcação.

3. Meronímia: Y é um merônimo de X se Y é parte de X, janela é um merônimo de prédio.

4. Holonímia é o inverso de meronímia, carro é um holônimo de pneu.

Na Figura 2.10 podem ser visualizadas algumas das relações disponíveis na WordNet a partir do substantivo car.

A WordNet é utilizada principalmente na etapa de seleção das respostas, na verificação se uma resposta candidata é do tipo esperado, pois isso é facilmente verificável

Figura 2.10: A navegação na hierarquia de hiperônimos da WordNet

através da hierarquia de hiperônimos da WordNet. A base de sinônimos da WordNet também é muito utilizada, pois permite que diversas consultas sejam feitas a partir da mesma pergunta, utilizando os sinônimos das palavras presentes na pergunta e recuperando documentos onde a resposta pode aparecer de uma forma diferente da pergunta. A falta de uma base léxica completa como a WordNet é uma grande limitação para o QA na língua portuguesa. Existe uma WordNetPt[22] (Portugal) (MARRAFA, 2002) e uma WordNetBr (Brasil) (SILVA, 2005), porém essas bases são muito limitadas, não contendo uma abrangência tão grande como a WordNet da língua inglesa, que contém mais de 166.000 pares (forma, sentido) de uma palavra, enquanto a WordNetPt contém aproximadamente 19.000 termos.

2.6.1.2 PAPEL

O PAPEL (Palavras Associadas Porto Editora Linguateca) (OLIVEIRA et al., 2008) é um conjunto de relações entre palavras da língua portuguesa. Esse recurso foi construído através de extracão semiautomática de padrões de expressões que ocorrem nas definições do Dicionário da Língua Portuguesa da Editora Porto. Dessa forma, foram identificadas relações composicionais, hierárquicas e de sinonímia. O PAPEL está sendo utilizado para o desenvolvimento de uma base léxica para o português (PRESTES et al., 2011), na forma de uma ontologia de domínio geral semelhante à WordNet. Alguns exemplos de relações do PAPEL podem ser vistos abaixo:

repartir SINONIMO_DE partilhar
vasqueiro PROPRIEDADE_DE_ALGO_QUE_CAUSA vasca
vazar ACCAO_QUE_CAUSA vazão

[22] WordNetPt, disponível em http://www.clul.ul.pt/clg/wordnetpt/index.html

cabo PARTE_DE vassoura
navio HIPERONIMO_DE veleiro

Devido a grande diferença de abrangência entre estes recursos para a língua portuguesa e para a língua inglesa, o sistema desenvolvido neste trabalho não utilizou ontologias para que a comparação entre as língua fosse equivalente.

2.6.2 Parsers

Um parser é um sistema que faz a análise sintática de um texto para determinar a sua estrutura gramatical. Um parser deve ter uma gramática para que ao analisar os textos de entrada ele reconheça se o texto pode ser derivado a partir desta gramática. Os parser modernos são estatíticos, utilizando textos manualmente anotados por especialistas para aprender a estrutura da linguagem utilizando técnicas como PCFGs (*probabilistic context free grammars*), máxima entropia ou redes neurais. Uma corpus anotado largamente utilizado para treinar e testar parser estatíticos para o inglês é o PennTreebank[23]. Também existem corpora anotados para o português como o Bosque[24], que faz parte do projeto Floresta Sintática da Linguateca[25]. Alguns exemplos de parsers são o Stanford parser (KLEIN; MANNING, 2003), o Berkeley parser [26], o RASP parser (BRISCOE; CARROLL; WATSON, 2006) para o inglês e o Palavras (BICK, 2000) para o português.

2.6.3 Reconhecedor de Entidades Nomeadas

Outro recurso muito importante em um sistema de QA é um reconhecedor de entidades nomeadas, o uso desses reconhecedores é importante para determinar as entidades do tipo de resposta esperado, uma etapa essencial para que o sistema responda corretamente à perguntas factóides. Um reconhecedor recebe um texto como entrada e marca as entidades que ele encontra, como na frase: *"Carl Barks nasceu em Merrill, Oregon."* serão identificadas as seguintes entidades: Carl Barks como pessoa, Merril como local e Oregon como local. Um reconhecedor de entidades utilizado em diversos sistemas é o BBNIdentiFinder (BIKEL; SCHWARTZ; WEISCHEDEL, 1999) com uma precisão de 93% e um recall de 96%. Porém é um sistema fechado desenvolvido por uma empresa. O parser Palavras também possui um reconhecedor de entidades nomeadas integrado ao analisador sintático, comentado na seção 3.2.2.

[23]PennTreebank, acessível em http://www.cis.upenn.edu/ treebank/
[24]Bosque, disponível em http://www.linguateca.pt/floresta/corpus.html#bosque
[25]Floresta Sintática, disponível em http://www.linguateca.pt/Floresta/
[26]Berkeley parser, disponível em http://code.google.com/p/berkeleyparser/

3 MATERIAIS E MÉTODOS

Neste capítulo serão descritos os recursos que foram utilizados neste trabalho. Primeiro descreveremos o corpus de perguntas utilizado, e depois as ferramenta: parser e reconhecedor de entidades nomeadas.

3.1 Corpus

As perguntas utilizadas para testes neste trabalho foram extraídas das trilhas de Perguntas e Respostas de 2004, 2005 e 2008 do CLEF. Essas perguntas estão disponiveis no repositório do QA@CLEF[1] e contam com anotações como: tipo de pergunta (*factoid, definition ou list*), tipo de resposta esperado, resposta e em alguns casos a frase de onde a resposta foi extraída. É importante que essas perguntas contenham as respostas para permitir a avaliação automática dos resultados, sem a necessidade da verificação manual de cada uma das perguntas. As perguntas estão distribuídas de acordo com as Tabelas 3.1 para as perguntas em português, e a Tabela 3.2 para o inglês:

Tabela 3.1: Distribuição das perguntas – português

	Factoid	Definiton	List	total
2004	608	92	0	700
2005	158	42	0	200
2008	162	28	10	200
total	928	162	10	1100

Tabela 3.2: Distribuição das perguntas – inglês

	Factoid	Definiton	List	total
2004	608	92	0	700
2005	150	50	0	200
2008	160	30	10	200
total	918	172	10	1100

[1] Repositório do CLEF, acessível em http://celct.fbk.eu/QA4MRE/index.php?page=Pages/pastCampaigns.php

É importante notar que somente as questões do CLEF do ano de 2004 são as mesmas para o português e o inglês, nos demais anos do CLEF, 2005 e 2008, as perguntas são diferentes para cada um dos idiomas. Nas Tabelas 3.3 e 3.4 são mostrados os tipos de resposta esperados para as perguntas em português e em inglês, respectivamente.

Tabela 3.3: Tipos de Resposta – português

	2004	2005	2008	**total**
PERSON	173	73	38	284
LOCATION	118	35	39	192
TIME	82	15	24	121
ORGANIZATION	98	38	16	152
MEASURE	84	18	15	117
MANNER	26	0	0	26
OBJECT	31	0	7	38
OTHER	88	21	42	151
COUNT	0	0	19	19
total	700	200	200	1100

Tabela 3.4: Tipos de Resposta – inglês

	2004	2005	2008	**total**
PERSON	173	55	30	258
LOCATION	118	22	21	161
TIME	82	20	20	122
ORGANIZATION	98	48	29	175
MEASURE	84	29	20	133
MANNER	26	0	0	26
OBJECT	31	0	29	60
OTHER	88	26	31	145
COUNT	0	0	20	20
total	700	200	200	1100

Neste trabalho focamos nas perguntas factóides, então em nossas avaliações foi utilizado somente esse tipo de pergunta. Como também necessitávamos de um método para a avaliação automática dos resultados de cada método de seleção da resposta, utilizamos somente as perguntas com respostas anotadas. Estas restrições nos deixaram com o número de perguntas válidas mostradas na Tabela 3.5:

Tabela 3.5: Questões válidas

	2004	2005	2008	**total**
português	588	144	152	884
inglês	588	133	160	881

3.2 Ferramentas

Nesta seção serão descritas as ferramentas utilizadas no sistema desenvolvido, o reconhecedor de entidades nomeadas de Stanford (FINKEL; GRENAGER; MANNING, 2005) e os parsers Palavras (BICK, 2000) e RASP parser (BRISCOE; CARROLL; WATSON, 2006).

3.2.1 Stanford NER

O reconhecimento de entidades nomeadas (*Named-Entity Recognition*, NER) busca identificar elementos em categorias pré-definidas como pessoas, organizações, expressões de tempo, quantidades, percentuais e outras. Um reconhecedor de entidades nomeadas é muito importante para um sistema de QA responder a perguntas factóides, isto é, que tem como resposta uma entidade, como as identificadas por um reconhecedor. A importância desse recurso se deve a sua capacidade de identificar a resposta exata dentro das frases extraídas pelo módulo de extração das respostas.

O Reconhecedor de Entidades nomeadas da universidade de Stanford (FINKEL; GRENAGER; MANNING, 2005) também foi desenvolvido baseado em um modelo de aprendizado de máquina, ele utiliza o Modelo de Máxima Entropia de Markov. O Stanford NER pode ser treinado, caso se disponha de um corpus com as as anotações de entidades. Neste trabalho utilizamos o Stanford NER com o modelo disponibilizado para reconhecimento de entidades na língua inglesa. Esse modelo foi treinado no CoNLL[2] 2003 e reconhece entidades dos tipos: pessoa, organização, local e variado(*miscelanious*). Na frase *The World Trade Center bombing occurred on February 26, 1993, when Ramzi Yousef ...*", são reconhecidas as entidades: Ramzi Yousef como pessoa, e World Trade Center como variado. O Stanford NER tem uma precisão de 86.12% e um recall de 86.49%.

3.2.2 Palavras

O Palavras (BICK, 2000) é um parser estatístico para a língua portuguesa. Ele é um parser robusto que sempre retorna uma análise mesmo para frases de entrada incompletas ou incorretas e tem uma baixa taxa de erros, menos de 1% para classes de palavras e entre 3 e 4% para sintaxe superficial.

Além da análise sintática da frase, o Palavras também conta com um reconhecedor de entidades nomeadas, descrito em (BICK, 2003) integrado ao parser. Durante a análise sintática, grande parte dos substantivos, verbos e alguns adjetivos são anotados com tags semânticas. Os tipos de entidades de nível mais alto reconhecidas pelo Palavras são nomes de pessoas, locais, organizações, eventos, produtos semânticos, como livros, filmes e músicas, e nomes de marcas e objetos, a documentação e descrição de outros tipos de entidades reconhecidas estão disponíveis em http://beta.visl.sdu.dk/visl/pt/info/portsymbol.html. O reconhecedor de entidade nomeadas do Palavras tem uma precisão de 91.8% e um *recall* de 91.9%.

Abaixo um exemplo de saída do Palavras:

```
Fernão=Capelo=Gaivota  [Fernão=Capelo=Gaivota] <hum> PROP M S @SUBJ>  #1->2
é  [ser] <vK> <fmc> <mv> V PR 3S IND VFIN @FS-STA  #2->0
um  [um] <arti> DET M S @>N  \#3->4
romance  [romance] <sem-r> N M S @<SC  \#4->2
de  [de] PRP @N<  \#5->4
```

[2]CoNLL, *Conference on Natural Language Learning*, site oficial acessível em http://ifarm.nl/signll/conll/

```
Richard=Bach  [Richard=Bach] <hum> PROP M S @P<  #6->5
, \#7->0
publicado  [publicar] <vH> V PCP M S @N<PRED  #8->6
em  [em] PRP @<ADVL  \#9->2
1970  [1970] <card> <date> <year> NUM M/F S @P<  #10->9
. \#11->0
```

Nela é possível ver as classes gramaticais das palavras (V, NUM, N PROP), as entidades reconhecidas (<hum>, <year>). A documentação do parser está disponível em http://beta.visl.sdu.dk/visl/pt/info/symbolset-manual.html e http://beta.visl.sdu.dk/visl/pt/info/portsymbol.html.

3.2.3 RASP Parser

O RASP Parser (BRISCOE; CARROLL; WATSON, 2006) é um parser estatístico para a língua inglesa. Ele tem quatro etapas em seu processamento, a primeira é a separação em *tokens*, que separa um texto de entrada em frases e separa as palavras da pontuação. A segunda etapa é o *part-of-speech tagging*, onde é inferido o PoS (classe gramatical) de cada palavra. A terceira etapa é a lematização, que é o processo de agrupar as diferentes formas flexionadas das palavras para que elas possam ser analisadas com um item único. A quarta e última etapa é o *parsing* probabilístico da sentença que gera a análise sintática final da frase.

Abaixo temos um exemplo de saída do Rasp Parser para a frase: *"Who was the creator of Tintim?"*

```
(|Who:1_PNQS| |be+ed:2_VBDZ| |the:3_AT| |creator:4_NN1| |of:5_IO| |Tintim:6_NP1|)
(|ncsubj| |be+ed:2\_VBDZ| |Who:1\_PNQS| \_)$
(|xcomp| _ |be+ed:2_VBDZ| |creator:4_NN1|)
(|det| |creator:4_NN1| |the:3_AT|)
(|iobj| |creator:4_NN1| |of:5_IO|)
(|dobj| |of:5_IO| |Tintim:6_NP1|)
```

Na primeira linha temos o PoS de cada palavra e nas demais linhas a análise sintática da frase, indicando relações gramaticais entre as palavras, descritas em (BRISCOE, 2006).

4 ARQUITETURA

Neste capítulo é descrito o sistema implementado para a avaliação dos métodos de seleção da resposta. Foram utilizados três sistemas de Perguntas e Respostas para a avaliação. O primeiro é um sistema de QA para o português, o segundo é um sistema de QA para o inglês, com a mesma arquitetura que o sistema para o português. As diferenças entre esses sistemas foram o parser e o reconhecedor de entidades nomeadas, que são elementos dependentes da língua utilizada. O terceiro sistema utilizado é o OpenEphyra, um sistema de QA complexo (que utiliza mais recursos linguísticos, como a WordNet) de código aberto. O OpenEphyra foi utilizado como comparativo aos outros dois sistemas desenvolvidos para verificar o impacto desses recursos na etapa de seleção da resposta.

4.1 Sistemas Desenvolvidos

Nesta seção é descrita a arquitetura dos dois sistemas desenvolvidos, QA para português e QA para inglês e é descrito as modificações e os módulos que foram utilizados do OpenEphyra.

4.1.1 Sistema de Perguntas e Respostas para o português

O sistema desenvolvido possui as quatro etapas básicas de um sistema de QA: análise da pergunta, busca dos documentos candidatos, geração das respostas candidatas e seleção da resposta.

Na análise da pergunta o mais importante é descobrir o tipo de resposta esperado. Para o sistema desenvolvido nesse estudo utilizamos uma versão adaptada para o português das *wh-phrases* (SRIHARI; LI, 2000). A Tabela 4.1 ilustra o mapeamento dos pronomes interrogativos do português para o tipo de resposta esperado. Esse mapeamento foi desenvolvido manualmente.

Como visto na Tabela 4.1, foram utilizados quatro tipos de resposta esperados: pessoa, local, número e tempo. O mapeamento utilizado é bem simples e de forma nenhum exaustivo, já que há inúmeras maneiras de se formular uma pergunta. Nota-se que na segunda linha da Tabela 4.1 o pronome quanto, tem diversas formas flexionadas, tanto em gênero, quanto em número, que devem ser levadas em conta para uma boa classificação do tipo de resposta esperado. Essa implementação mais simples, com poucos tipos e mais abrangentes, da etapa de análise da pergunta foi escolhida para que não ocorram erros de classificação errada nessa etapa, prejudicando as etapas seguintes.

Na etapa de Busca dos Documentos Candidatos, também optamos por uma abordagem mais simples. Na formulação da consulta, removemos as *stopwords* da pergunta e enviamos uma consulta ao Google, identificamos os primeiros dez documentos retornados

Tabela 4.1: Regras utilizadas para o tipos de resposta esperado no português

PRONOME	TIPO DE RESPOSTA ESPERADO
quem	PESSOA
qual nome	PESSOA
quanto(s)(a)(as)	NUMERO
quando	TEMPO
que ano	TEMPO
onde	LUGAR
(em) que cidade	LUGAR
(em) que país	LUGAR

e extraímos os textos deles para a análise.

Na geração das respostas candidatas, os textos retornados na etapa anterior são divididos em frases, e buscamos, em cada frase, alguma palavra em comum com as palavras da pergunta. Todas as frases com alguma palavra em comum são analisadas pelo Palavras e todas as entidades nomeadas são extraídas de cada frase. As entidades do tipo de resposta esperada, obtidas na etapa de análise da pergunta são salvas em pares (entidade, frase). Caso o tipo esperado de resposta não tenha sido identificado, também são extraídos todos os substantivos das frases e armazenados junto com as frases em que eles ocorrem. Isso é feito para que se possa responder a perguntas como *"Como se chamam os pilotos suicidas japoneses?"*, cuja resposta kamikazes não é reconhecida como uma entidade. O Palavras possui um reconhecedor de entidades nomeadas com diversas categorias como visto na Seção 3.2.2, porém elas tiveram que ser adaptadas para se adequar ao módulo de análise da pergunta e os tipos de resposta esperado que ele reconhece. Assim temos os mesmo quatro tipos de resposta esperados reconhecidos como entidades pelo Palavras: Pessoa, Local, Número e Data.

A Seleção das Respostas tem como entrada uma lista de pares (frase, resposta) e tem como saída essa lista, ordenada de acordo com um score de confiança atribuído a cada par (frase, resposta), que quanto mais alto, mais provável de que a resposta esteja certa. Essa etapa é o alvo do estudo desse trabalho e diversas abordagens foram implementadas e testadas, cada uma dessas técnicas de seleção da resposta será detalhada na seção 4.2

4.1.2 Sistema de Perguntas e Respostas para o inglês

O sistema desenvolvido para o inglês tem a mesma estrutura básica que o sistema para o português, descrito na seção anterior. Aqui descreveremos apenas as mudanças necessárias de recursos dependentes de linguagem.

Assim como no português, na etapa de análise da pergunta utilizamos um mapeamento simples dos pronomes e expressões interrogativas para os tipos de resposta esperado. O mapeamento utilizado está na Tabela 4.2 e foi adaptado para ser semelhante ao utilizado para o português.

A busca dos documentos candidatos para o inglês tem como diferenças do português a lista de *stopwords* que serão eliminadas antes de submeter a consulta ao Google e a especificação da língua para retornar páginas em inglês. Os textos dos dez primeiros documentos retornados são passados ao próximo módulo.

Na geração das respostas candidatas são selecionadas as frases com alguma palavra em comum com a pergunta, e essas frases são analisadas com o Reconhecedor de Entida-

Tabela 4.2: Regras utilizadas para o tipos de resposta esperado no inglês

PRONOME	TIPO DE RESPOSTA ESPERADO
who	PERSON
how many	NUMBER
how much	NUMBER
how old	NUMBER
when	TIME
what/which year	TIME
which decade	TIME
which date	TIME
what time	TIME
where	LOCATION
what/which city	LOCATION
what/which country	LOCATION
what/which town	LOCATION

des Nomeadas de Stanford, e caso o tipo da resposta esperado não tenha sido identificado, a frase é analisada com o RASP parser e os substantivos são extraídos, para que perguntas como *"What animal do players ride in the game of polo?"*, cuja resposta horse não é reconhecida como uma entidade. No final desta etapa é gerada uma lista de pares (entidade, frase) que serão analisados pelos algoritmos de seleção da resposta.

A etapa de seleção da resposta não sofreu alterações em relação ao português e está descrita na seção 4.2.

4.1.3 Sistema Estendido de Perguntas e Respostas para o inglês

O terceiro sistema utilizado foi o OpenEphyra, descrito na Seção 2.3.1.6. Os componentes utilizados foram os módulos completos de análise a pergunta, geração da consulta e busca. No módulo de extração e seleção da resposta foram utilizados somente os filtros de extração das respostas que são: *answer pattern filter, answer type filter, semantic extraction filter* e *answer projection filter*, que podem ser visualizados na Figura 2.7. Após a utilização desses componentes do OpenEphyra, aplicamos os algoritmos de seleção da resposta descritos na Seção 4.2 às respostas candidatas retornadas pelo OpenEphyra.

4.2 Algoritmos de Seleção da Resposta

Nesta seção serão descritos todos os algoritmos de seleção da resposta utilizados neste trabalho. Os seguintes algoritmos serão testados: *bag-of-words*, *n-grams*, distância *keywords*, Google score, Wikipedia score bem como a utilização de algoritmos de aprendizado de máquina J48 e SVM. Estes algoritmos foram escolhidos por que são alguns dos frequentemente utilizados nos trabalhos da área e cobrem diversos tipos de abordagens para a seleção da resposta, desde os algoritmos mais superficiais como o *bag-of-words* até algoritmos que envolvem aprendizado de máquina.

4.2.1 Similaridade *Bag-of-Words*

Bag-of-words, descrito em (QUARTERONI, 2007), é um dos algoritmos mais simples para a seleção da resposta. Ele atribui um score para uma resposta candidatas de acordo com o número de palavras em comum com a pergunta. Este score representa a similaridade entre a pergunta e a resposta candidata. Formalmente o score *bag-of-words* é calculado de acordo com a Equação 4.1:

$$bow(p, r) = \sum_{i<|r|,j<|p|} match(r_i, p_j) \tag{4.1}$$

Onde p é a pergunta, r é a frase de resposta, $|r|$ é tamanho da frase de resposta, $|p|$ o tamanho da pergunta e $match(a, b)$ é uma função que recebe duas sequências de caracteres como parâmetro e retorna 1 caso eles sejam iguais e 0 caso sejam diferentes.

Por exemplo, caso tenhamos a resposta *"Fuji"*, encontrada na frase *"A montanha mais alta no Japão, Fuji é o símbolo mais familiar do país"*, para a pergunta *"Qual a montanha mais alta do Japão?"* o score *bag-of-words* será igual a 4 pois teremos 4 palavras em comum: *"montanha", "mais", "alta" e "Japão"*.

4.2.2 N-Grams

A similaridade de *n-grams* estudada neste trabalho, também descrita em (QUARTERONI, 2007), é utilizada quando a simples correspondência de palavras não é o suficiente para que uma das respostas seja escolhida, a similaridade de *n-grams* estabele uma similaridade mais forte entre a pergunta e resposta candidata, ela verifica se as palavras estão na mesma ordem em ambas as frase e agrupadas. A Equação 4.2 descreve como é feito o cálculo da similaridade de *n-gramas*

$$ng(p, r) = \sum_{i<|r|,j<|p|} match\_n(r_i, p_j, n) \tag{4.2}$$

Onde $match\_n(r_i, p_j, n)$ é uma função que recebe dois elementos de duas listas de sequências de caracteres(r_i e p_j) e um inteiro (n) e retorna 1 caso os n elementos das duas listas começando em i e j sejam iguais, por exemplo, caso $n = 3$, para que $match\_n$ retorne 1 temos que ter $r_i = p_j$, $r_{i+1} = p_{j+1}$ e $r_{i+2} = p_{j+2}$. Neste trabalho utilizamos $n = 2$, bigramas, assim como em (QUARTERONI, 2007). Também fizemos alguns testes com $n = 3$, porém a grande maioria dos resultados foram iguais a zero, indicando que não há quase nenhuma similaridade de trigramas entre as respostas candidatas e as perguntas.

Seguindo o mesmo exemplo apresentado na Seção 4.2.1, a resposta *"Fuji"* terá um score *n-gram* igual a 2 (utilizando $n = 2$) pois teremos dois bigramas correspondentes: *"montanha mais"* e *"mais alta"*.

4.2.3 Distância *Keywords*

A distância das *keywords* é a métrica utilizada na seleção da resposta pelo MULDER, detalhado na seção 2.3.1.1. Neste algoritmo o score atribuído a uma resposta candidata é dado pela distância entre a resposta e as *keywords*[1] da pergunta, que neste trabalho são as palavras que restam após a eliminação das *stopwords*. Formalmente, sejam k_i as palavras chave das respostas, a_i as palavras da resposta e c_i outras palavras. Suponha que temos

[1] Keywords ou palavras chave são as palavras mais importantes em uma pergunta, elas podem ser obtidas com uma análise superficial como obter as palavras que restam após a eliminação de *stopwords* ou uma análise mais profunda utilizando informações sintáticas e semânticas obtidas com um parser, como extrair somente palavras de alguma classe gramatical

uma lista de palavras chave consecutivas à esquerda da resposta candidata separadas por m palavras não relacionadas, isto é, $k_1k_2...k_nc_1c_2...c_ma_1a_2...a_p$, o score é computado de acordo com a Equação 4.3:

$$K_{esq} = \frac{n}{m+1} \tag{4.3}$$

A divisão é feita sobre $m+1$‘ para evitar divisão por zero. A mesma equação é computada para as palavras à direita da resposta candidata na frase que a contém, K_{dir}, e o maior score entre K_{dir} e K_{esq} é escolhido como o score da resposta candidata.

Continuando com o exemplo, temos um score utilizando distância de *keywords* igual a 1 pois temos o maior valor na esquerda onde *Japão* está ao lado da resposta, portanto existem 0 palavras não relacionadas separando-as.

4.2.4 Google Score

Baseado em (MAGNINI et al., 2002) o Google é usado para gerar um score numérico. Uma consulta consistindo da resposta candidata e das palavras chave da pergunta é enviada ao Google, e os 10 primeiros *snippets*[2] retornados são analisados utilizando o Algoritmo 1, que calcula a distância entre as palavras chave da pergunta e a resposta candidata.

Algoritmo 1 Score de validação da resposta com o Google

Entrada: Uma lista de respostas candidatas $A_1, A_2, ..., A_n$.
Saida: Um score de validação para cada resposta candidata.

Para cada resposta candidata A_i

1. Inicializa o Google score: gs(A_i) = 0
2. Para cada snippet s:
   2.1 Inicializa o score de co-ocorrência do snippet: $cs(s) = 1$;
   2.2 Para cada palavra chave k em s:
   2.2.1 Computa a distância d, o número mínimo de palavras entre k e a resposta candidata
   2.2.2 Atualiza o score de co-ocorrência do snippet: $cs(s) = cs(s) \times 2^{(1+d)^{-1}}$
   2.3 $gs(A_i) = gs(A_i) + cs(s)$

4.2.5 Tf-Idf

O tf-idf não foi utilizado como algoritmo de seleção da resposta neste estudo, mas ele é parte importante do algoritmo Wikipedia score detalhado na seção 4.2.6. O tf-idf é uma métrica muito utilizada em recuperação de informação. Ele é utilizado para medir o quão importante é uma palavra dentro de um corpus (uma coleção de documentos). Essa importância cresce proporcionalmente ao número de vezes que essa palavra aparece no documento e é influenciada pela frequência da palavra no corpus. Variações do tf-idf são muito utilizadas em ferramentas de busca para ordenar documentos dada uma consulta. O tf, *term frequency* é simplesmente o número de vezes que uma palavra aparece em um documento: O idf, *inverse document frequency* é uma medida da importância do termo. Ele é calculado dividindo o número de documentos no corpus pelo número de documentos

[2]Snippets são trechos de texto resumindo a página apontada

que possuem o termo e calculando o logaritmo desse quociente.

$$idf(t) = log\frac{|D|}{|d : t \in d|} \tag{4.4}$$

onde $|D|$ é o número de documentos na coleção e $|d : t \in d|$ é o número de documentos onde t aparece.

Para obter o idf das palavras, necessário para se calcular o score de validação da Wikipedia, é necessário saber o número de documentos na coleção e a quantidade de documentos dessa coleção que contém a palavra procurada. Como estamos avaliando um sistema de Perguntas e Respostas que usa a internet como base de dados para buscar suas informações, é uma tarefa colossal medir o número de documentos presentes na web e contar quantos destes contém cada palavra.

Para fazer uma aproximação do idf, utilizamos o corpus WikiXML[3] para calcular o idf das palavras presentes na Wikipedia. Os arquivos do WikiXML da versão em português da Wikipedia continham 312.170 arquivos totalizando 3,2 Gbytes de arquivos, enquanto os arquivos utilizados para aproximar o idf das palavras na Wikipedia versão inglesa continham 297.267 arquivos totalizando 5,0 Gbytes.

4.2.6 Wikipedia Score

O algoritmo para validação da resposta utilizando a Wikipedia[4] é mostrado no algoritmo 2. Ele consiste em procurar um documento na Wikipedia cujo título é a resposta candidata e analisá-lo para obter a frequência do termo (tf) e a frequência inversa do documento (idf) e assim calcular o tf-idf. Quando não há documento com o título da resposta candidata cada palavra chave da pergunta é processada da mesma maneira, procura-se um documento cujo título é a palavra e calcula-se o tf-idf da resposta candidata naquele documento.

Algoritmo 2 Score de validação da resposta com a Wikipedia

Entrada: Uma lista de respostas candidatas $A_1, A_2, ..., A_n$.
Saida: Um score de validação para cada resposta candidata.

Para cada resposta candidata A_i

1. Inicializa o Wikipedia score: $ws(A_i) = 0$
2. Procura na Wikipedia um documento cujo título é A_i.
3. Se encontrou, calcula o tf-idf de A_i no documento
   $ws(A_i) = (1 + log(tf)) \times (1 + log(idf))$
4. Se não encontrou, para cada palavra chave da pergunta K_j
   4.1 Procura na Wikipedia por um documento cujo título seja K_j
   4.2 Se encontrou o documento, calcula o tf-idf de A_i:
   $ws(A_i) += (1 + log(tf)) \times (1 + log(idf))$

4.2.7 Algoritmos de Aprendizado de Máquina

Algoritmos de aprendizado de máquina são muito utilizados em sistemas de Perguntas e Respostas, principalmente na parte de análise da pergunta e na parte de seleção da resposta, como visto em 2.4.4, em geral esses algoritmos são treinados para classificar o

[3] WikiXML, disponível em http://ilps.science.uva.nl/WikiXML/
[4] Wikipedia, acessível em www.wikipedia.org

tipo de pergunta a partir de um corpus anotado. Para o estudo dos algoritmos de aprendizado de máquina na seleção da resposta utilizamos o WEKA[5]. O WEKA é um software de código aberto desenvolvido na Universidade de Wakaito, Nova Zelândia que agrega diversos algoritmos de aprendizado de máquina de vários paradigmas dentro da Inteligência Artificial, como, por exemplo, árvores de decisão e algoritmos bayesianos.

Neste trabalho utilizaremos os seguintes algoritmos de aprendizado de máquina para a seleção da resposta: (1) J48, e (2) SVM. Os algoritmos de aprendizado de máquina recebem um conjunto de dados para treino e com eles "aprendem" um modelo de classificação, no caso da seleção da resposta queremos classificar as respostas em corretas e erradas. O algoritmo J48 utiliza o conjunto de dados de treino para montar uma árvore de decisão que a partir dos valores de um dado de entrada classifica a resposta. O algoritmo SVM é modela os dados como pontos em um espaço n-dimensional (n é o número de atributos de um dado de entrada) e busca traçar um hiperplano que separa esses valores em corretos e incorretos. Para todos os algoritmos testados utilizamos como atributos das respostas os valores obtidos nos algoritmos anteriores, ou seja, *bag-of-words, n-gramas*, distância entre *keywords*, Google score e Wikipedia score. Como um exemplo de vetor de entrada para a resposta *"Fuji"* temos os seguintes valores: 4, 2, 1, 71.46, 29.49 e true pois ela é uma resposta correta. Para uma resposta incorreta temos os valores calculados e false indicando que ela é incorreta.

4.2.8 Agrupamento de Respostas

Na extração das respostas da web ou de documentos encontramos diversas respostas iguais ou equivalentes, que se referem à mesma entidade, como, por exemplo, a lista de resposta candidatas para a pergunta *"Quem foi a primeira mulher no espaço?"*, inclui Valentina Vladimirovna Tereshkova e Valentina Tereshkova, que se referem a mesma pessoa e estão ambas corretas. Sendo assim, criamos um método para agrupar essas respostas equivalentes e explorar essa característica da extração das respostas.

Para o estudo do impacto do agrupamento das respostas na seleção da resposta, implementamos um algoritmo simples para o agrupamento das respostas: Consideramos equivalentes as respostas quando uma delas contém as palavras da outra. Nesse caso juntamos as duas respostas em uma só e somamos seus scores calculados anteriormente (*bag-of-words, n-gramas*, distância *keywords*, Google score e Wikipedia score). Os scores são somados para que estas respostas agrupadas tenham uma maior chance de serem selecionadas. A resposta com os maiores scores permanece como a representante do grupo de respostas. Avaliaremos o impacto do agrupamento de respostas para cada um dos algoritmos de seleção da resposta, exceto os algoritmos de aprendizado de máquina.

[5]WEKA, disponível em http://www.cs.waikato.ac.nz/ml/weka/

5 AVALIAÇÃO

Nesta capítulo iremos descrever a metodologia de testes e avaliação utilizadas para a comparação dos resultados dos diversos algoritmos de seleção da resposta.

5.1 Primeira Etapa: Extração de Respostas

A primeira etapa da metodologia de avaliação consistiu em rodar os três sistemas até o final da etapa de geração das respostas candidatas e armazená-las para posterior avaliação dos algoritmos de seleção da resposta. A entrada de cada sistema consiste na pergunta, sua resposta (utilizada para avaliação automática dos resultados, mas não usada nos experimentos) e a saída consiste em uma lista de pares de respostas candidatas e frase contendo essa resposta. Na Tabela 5.1 temos os resultados da primeira etapa, o número em cada célula significa o percentual de perguntas que tiveram a resposta correta extraída entre todas as respostas candidatas presentes ao final da etapa de geração das respostas candidatas.

Tabela 5.1: Resultados da etapa 1 do experimento: a geração das respostas candidatas

	2004	2005	2008	**média**
português	68,0%	63,8%	75,0%	68,9%
inglês	75,0%	71,4%	73,1%	**73,1%**
ephyra	62,1%	42,8%	50,6%	51,8%

Como podemos verificar na Tabela 5.1 o sistema com melhor desempenho foi o QA para o inglês, os resultados do QA para português ficaram um pouco abaixo, mostrando que há alguma diferença na informação disponível na web para as duas línguas ou então nos parsers e reconhecedores de entidades nomeadas utilizados no final da etapa de geração das repostas candidatas. Um resultado que também chama a atenção foi o desempenho bem abaixo dos outros sistemas do OpenEphyra.

5.2 Segunda Etapa: Seleção da Resposta

A segunda etapa consistiu em utilizar os algoritmos de seleção da resposta estudados nas respostas extraídas na primeira etapa.

5.2.1 Taxa de acerto para cada sistema

Na Tabela 5.2 podemos ver as taxas de acertos dos algoritmos de seleção da resposta estudados para cada um dos idiomas. Esta taxa foi calculada utilizando todas as perguntas válidas utilizadas no experimento descritas na Tabela 3.5. Esse percentual se refere a quantidade de respostas certas escolhidas (rankeadas em primeiro lugar) pelo algoritmo dividido pelo número de perguntas com respostas certas extraídas na primeira etapa.

Tabela 5.2: Resultados etapa 2 – seleção da resposta

	bow	ngram	dist key	Google	Wikipedia
português	15,8%	14,9%	**16,0%**	3,9%	12,6%
inglês	16,6%	16,2%	**18,6%**	3,0%	11,4%
ephyra	17,5%	17,5%	16,9%	2,2%	**24,9%**

Na Tabela 5.3 temos o percentual de respostas corretas, calculado da mesma maneira para cada um dos algoritmos em cada idioma, após o agrupamento de respostas.

Tabela 5.3: Resultados etapa 2 – Agrupamento

	bow	ngram	dist key	Google	Wikipedia
português	**31,8%**	30,5%	30,1%	26,6%	28,6%
inglês	**30,7%**	28,3%	30,0%	23,6%	28,1%
ephyra	25,2%	23,3%	23,3%	**55,2%**	35,4%

Podemos notar que o desempenho de todos os algoritmos é melhorado com o agrupamento das respostas, o que indica que temos uma redundância de informações na web que nos ajuda a encontrar as informações corretas e distinguí-las das incorretas. Os resultados dos sistemas para português e para inglês foram semelhantes em todos os aspectos, porém os mesmos algoritmos de seleção da resposta aplicados às respostas candidatas extraídas pelo OpenEphyra tiveram um desempenho bem diferente em alguns casos, como o Wikipedia score sem agrupamento, que teve um desempenho bem melhor e o Google score, que apesar do fraco desempenho no caso normal, teve um desempenho surpreendente com agrupamento, com uma taxa de acerto de 55,2% com agrupamento contra 2,2% sem.

5.2.1.1 Desempenho dos Algoritmos de Aprendizado de máquina

Para testar se um classificador influencia a performance da seleção utilizamos o WEKA com validação cruzada 10 vezes. O conjunto de dados inclui 1154 respostas candidatas, anotadas com os valores dos algoritmos estudados (*bag-of-words, n-grams*, distância *keywords*, Google score e Wikipedia score) e um campo indicando se é uma resposta correta ou não. O conjunto de dados foi balanceado incluindo aproximadamente metade de respostas corretas e metade incorretas, 581 corretas e 573 incorretas.

Os resultados foram medidos utilizando precisão, *recall* e *f-measure*, medidas tradicionais em recuperação de informação. Precisão é definida segundo a Equação 5.1, é a fração dos elementos recuperados que são relevantes.

$$p = \frac{|elementos\ relevantes| \cap |elementos\ recuperados|}{|elementos\ recuperados|} \tag{5.1}$$

Recall é definido pela equação 5.2, é a fração dos elementos relevantes que foram recuperados.

$$r = \frac{|elementos\ relevantes| \cap |elementos\ recuperados|}{|elementos\ relevantes|} \tag{5.2}$$

F-measure[1] é a média harmônica da precisão e do *recall* e é definido pela equação 5.3

$$f\text{-}measure = 2 \cdot \frac{p \cdot r}{p + r} \tag{5.3}$$

Os resultados para o português estão na Tabela 5.4 e para o inglês na Tabela 5.5:

Tabela 5.4: Resultados dos algoritmos de aprendizado de máquina – português

	precisão	*recall*	*f-measure*
J48	0,599	0,597	0,596
SVM	0,501	0,501	0,452

Tabela 5.5: Resultados dos algoritmos de aprendizado de máquina – inglês

	precisão	*recall*	*f-measure*
J48	0,608	0,608	0,608
SVM	0,500	0,501	0,496

Como podemos observar nas tabelas os algoritmos de aprendizado de máquina tem uma performance melhor que a de qualquer outro algoritmo de seleção de resposta testado, mesmo sendo treinados com características superficiais das frases. Os resultados mostram que o algoritmo de árvores de decisão obteve um desempenho levemente superior ao SVM. Também é possível observar que nas Tabelas 5.4 e 5.5 que os resultados para os dois idiomas são muito semelhantes.

5.2.2 Comparação por Idioma

A comparação dos sistemas por idiomas foi feita somente com as questões do CLEF do ano de 2004, que são idênticas para os dois idiomas. Essa comparação é feita com os sistemas português e inglês. Assim como nas tabelas anteriores esse resultado se refere ao número de perguntas certas dividido pelo número de perguntas com resposta certa extraídas na primeira etapa.

Tabela 5.6: Resultados português x Resultados inglês

	português	inglês	português c/ agrup.	inglês c/ agrup.
Bag-of-words	**16,5%**	15,1%	**34,0%**	**31,8%**
N-gramas	15,1%	15,4%	31,7%	27,8%
Distância *Keywords*	14,8%	**17,9%**	30,0%	30,5%
Google Score	3,4%	2,4%	28,5%	24,1%
Wikipedia Score	11,7%	11,9%	29,7%	29,1%

[1] *F-measure*, também chamada de *F-score* ou F_1 *score*

Comparando dois sistemas com a mesma arquitetura para dois idiomas diferentes observamos que não há diferenças significativas no desempenho dos algoritmos de seleção da resposta estudados.

5.2.3 Comparação por recursos utilizados no sistema

Nesta seção compararemos os resultados de cada algoritmo para os sistemas simplificado para inglês e o OpenEphyra utilizando os algoritmos estudados de seleção da resposta. A Tabela 5.7 mostra os resultados desses dois sistemas com e sem agrupamento de respostas.

Tabela 5.7: Resultados sistema para inglês x Resultados OpenEphyra

	inglês	ephyra	inglês c/agrupamento	ephyra c/agrupamento
Bag-of-words	15,1%	17,5%	**31,8%**	25,2%
N-gramas	15,4%	17,5%	27,8%	23,3%
Distância Keywords	**17,9%**	16,9%	30,5%	23,3%
Google Score	2,4%	2,2%	24,1%	**55,2%**
Wikipedia Score	11,9%	**24,9%**	29,1%	35,4%

Comparando os sistemas para o mesmo idioma porém com arquiteturas diferentes podemos observar grandes diferenças de desempenho do OpenEphyra nos algoritmos Wikipedia score sem agrupamento e Google score com agrupamento, que obteve o melhor desempenho entre todos os algoritmos estudados. Esse fato é mais surpreendente pelo fato desse algoritmo ter o pior desempenho em todos os casos sem agrupamento. Podemos verificar que a arquitetura diferente nas etapas iniciais de um QA pode modificar os resultados da seleção da resposta, pois eles foram diferentes para alguns algoritmos, mas muito semelhantes em outros.

5.2.4 Resultados dos sistemas completos

Os resultados dos sistemas como um todo, incluindo todas as etapas de um QA estão descritos na Tabela 5.8. Esses resultados foram obtidos utilizando todas as perguntas válidas do corpus e calculando o percentual de corretas em cada sistema avaliado. Nas três primeiras linhas da tabela estão os resultados dos sistemas utilizando os melhores algoritmos de seleção da resposta avaliados. A quarta linha da tabela contém os resultados do OpenEphyra rodando com seus próprios algoritmos de seleção da resposta.

Tabela 5.8: Resultados dos sistemas completos

	melhor algoritmo	melhor algoritmo com agrupamento
português	11,0%	21,9%
inglês	13,6%	22,4%
ephyra	12,9%	28,6%
ephyra completo	18,2%	não se aplica

Nessa tabela podemos observar que um QA utilizando estes algoritmos de seleção da resposta sem agrupar as respostas equivalentes tem um desempenho muito fraco. Mesmo com um melhor desempenho com o agrupamento, ultrapassando o OpenEphyra, este continua sendo muito abaixo dos melhores sistemas, como o Priberam (AMARAL et al., 2005) que tem um desempenho de mais de 64% de acertos.

5.3 Análise dos resultados

Os resultados apresentados pelos diversos algoritmos estudados nos mostram que a tarefa de seleção da resposta é realmente muito desafiadora. Todos algoritmos tem um desempenho muito fraco, tanto os mais simples como *bag-of-words* e *n-grams*, como os mais elaborados como o Google score e Wikipedia score que obtiveram resultados decepcionantes. O agrupamento das respostas mostrou ser uma boa alternativa para a seleção da resposta, sendo que ela aumentou o percentual de respostas corretas em todos os casos.

Um dos fatores que contribuiu para o fraco desempenho dos algoritmos de seleção foi a análise da questão muito simplificada e a extração de todos os substantivos das frases quando não era encontrado um tipo de resposta esperado. Isso aumentou as chances de ter a resposta correta entre as candidatas, porém a quantidade muito grande de respostas candidatas tornou a tarefa de seleção da resposta muito mais difícil.

É importante notar que existem algumas perguntas nessa base que são extremamente dependentes de tempo, ou seja, suas respostas estão atreladas à época em que a pergunta foi feita, ou então são dependendes do corpus de pesquisa que os sistemas tem acesso durante o CLEF. Como exemplo temos a pergunta número 504 de 2004, *"Onde ocorreu uma catástrofe ecológica?"*, essa pergunta tem como resposta *"Schweizerhalle, perto de Basileia"*, que provavelmente ocorreu à época ou então é a única catástrofe ecológica mencionada no corpus. O sistema desenvolvido encontrou como prováveis respostas para esta mesma pergunta, Teresópolis ou Rio de Janeiro, se referindo às catástrofes ocorridas no início de 2011. Outro exemplo de pergunta com depedência da época em que ela é feita é *"Quem interpreta o papel de James Bond no último filme da série 007?"* cuja resposta indicada como correta é Pierce Brosnan, o que já não é mais verdade.

Também foram encontradas perguntas como por exemplo a número 540, de 2004, *"Há quantos anos funciona o Greenpeace?"*, que obviamente, hoje tem uma resposta acrescida de 7 anos da que tinha em 2004. Outro fato importante de ressaltar é que a grande maioria das perguntas e suas respostas estão em português europeu, o que não acarreta em grandes problemas, mas pode gerar erros na avaliação automática utilizada neste trabalho. Como exemplo podemos citar a pergunta *"Onde fica o oásis de Siwa?"*, que tem como resposta *"Egipto"* que a grafia do país para português europeu. Nesta pergunta verificamos que a resposta com maiores scores foi justamente *"Egito"*, mas usando a grafia para o português brasileiro.

6 CONCLUSÃO

Os sistemas de Perguntas e Respostas podem diminuir o esforço dos usuários ao fazerem consultas simples à web. Porém para isso eles precisam ter suas quatro etapas bem desenvolvidas, em especial a etapa de seleção da resposta, que é a etapa final de um sistema de QA e a que mais distingue um sistema desse tipo de um buscador. Neste trabalho realizamos uma comparação entre métodos de seleção da resposta em sistemas de Perguntas e Respostas. Esses métodos foram comparados entre um sistema que utiliza a mesma arquitetura para o português e para o inglês e um sistema estendido para o inglês. Essa comparação foi feita para verificar o impacto de cada um desses métodos nas diferentes línguas e verificar se podemos ter um desempenho na língua portuguesa comparável ao da língua inglesa que possui mais recursos para a construção de um sistema de QA. Para a avaliação dos métodos construímos dois sistemas, um para o português, e outro para o inglês, com as quatro etapas básicas de um QA: análise da pergunta, busca dos documentos candidatos, extração das respostas e finalmente a seleção da resposta. Na seleção da resposta foram comparados os algoritmos: *bag-of-words, n-gramas*, distância entre *keywords*, Google score e Wikipedia score. Também foi comparado o desempenho de todos esses algoritmos após o agrupamento das respostas equivalentes. Por fim comparamos o desempenho de algoritmos de aprendizado de máquina para a seleção da resposta. Esses algoritmos utilizaram os valores calculados pelos algoritmos anteriores para classificar as respostas em corretas e incorretas.

Na avaliação dos métodos verificamos um desempenho bem abaixo do esperado dos algoritmos Google score e Wikipedia score nos sistemas simplificados, indicando que esses scores de validação não tem grande valor para a seleção da resposta se utilizados de maneira isolada ou em um sistema muito simples. Os melhores métodos de extração foram o *bag-of-words* e o distância *keywords*. O OpenEphyra apresentou um comportamento diferente dos outros sistemas com os algoritmos Wikipedia score e Google score Também verificamos que em todos os casos o agrupamento das respostas aumenta o número correto de respostas.

Comparando os sistemas por língua, todos os algoritmos de seleção de resposta tiveram um resultados semelhantes para o português e para o inglês. Já na comparação por complexidade do sistema, o OpenEphyra utilizando os algoritmos de seleção da resposta Wikipedia score sem agrupamento e Google score com agrupamento obtiveram resultados bem melhores que no sistema estendido para inglês mostrando que as primeiras etapas de um QA podem influenciar muito na seleção da resposta, como por exemplo o grande ganho de performance do Google score. Não obtivemos um bom desempenho em nenhum dos sistemas completos avaliados, nem mesmo com o agrupamento de respostas que aumentou o desempenho de todos os algoritmos de seleção da resposta.

O melhor algoritmo de seleção da resposta para sistemas simples é o algoritmo mais

simples, o Bag-of-Words, e o melhor algoritmo estudado para sistemas estendidos como o OpenEphyra é o Google score, porém ele é extremamente dependente de explorar a similaridade das respostas candidatas utilizando agrupamento de resposta, caso contrário seu desempenho é o pior dentre os algoritmos estudados.

Como trabalhos futuros pretendemos estender a avaliação para as outras etapas do processo de Question Answering, desenvolvendo uma taxonomia mais completa e mais abrangente para que a quantidade de respostas candidatas seja pequena e o desempenho dos algoritmos de seleção da resposta melhore. Também pretendemos estudar maneiras mais profundas de agrupar as respostas, visto que esse se mostrou um método que contribui muito para aumentar a taxa de acertos de um QA, assim como, estudar abordagens para melhorar os resultados da recuperação de documentos.

REFERÊNCIAS

ACCURATE UNLEXICALIZED PARSING, Stroudsburg, PA, USA. **Anais...** Association for Computational Linguistics, 2003.

AMARAL, C. et al. A Workbench for Developing Natural Language Processing Tools. In: IN PRE-PROCEEDINGS OF THE 1ST WORKSHOP ON INTERNATIONAL PROOFING TOOLS AND LANGUAGE TECHNOLOGIES (PATRAS, GREECE. **Anais...** [S.l.: s.n.], 2004. p.1–2.

AMARAL, C. et al. Priberam's question answering system for Portuguese. In: **Anais...** [S.l.: s.n.], 2005.

BAKER, C. F.; FILLMORE, C. J.; LOWE, J. B. The Berkeley FrameNet Project. In: COMPUTATIONAL LINGUISTICS - VOLUME 1, 17., Stroudsburg, PA, USA. **Proceedings...** Association for Computational Linguistics, 1998. p.86–90. (COLING '98).

BICK, E. **The Parsing System "Palavras". Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework**. [S.l.]: University of Arhus, 2000.

BICK, E. Multi-level NER for Portuguese in a CG framework. In: COMPUTATIONAL PROCESSING OF THE PORTUGUESE LANGUAGE, 6., Berlin, Heidelberg. **Proceedings...** Springer-Verlag, 2003. p.118–125. (PROPOR'03).

BIKEL, D. M.; SCHWARTZ, R.; WEISCHEDEL, R. M. **An Algorithm that Learns What's in a Name**. 1999.

BRISCOE, T. **An introduction to tag sequence grammars and the RASP system parser**. [S.l.]: University of Cambridge, Computer Laboratory, 2006. (UCAM-CL-TR-662).

BRISCOE, T.; CARROLL, J.; WATSON, R. The second release of the RASP system. In: COLING/ACL ON INTERACTIVE PRESENTATION SESSIONS, Stroudsburg, PA, USA. **Proceedings...** Association for Computational Linguistics, 2006. p.77–80. (COLING-ACL '06).

BURGER, J. et al. **Issues, Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A)**. [S.l.]: NIST, 2001.

CARVALHO, G.; MATOS, D. M. de; ROCIO, V. Improving IdSay: a characterization of strengths and weaknesses in question answering systems for portuguese. In: PROPOR. **Anais...** Springer, 2010. p.1–10. (Lecture Notes in Computer Science, v.6001).

CORTES, C.; VAPNIK, V. Support-Vector Networks. In: MACHINE LEARNING. **Anais...** [S.l.: s.n.], 1995. p.273–297.

DANG, H. T.; KELLY, D.; LIN, J. J. Overview of the TREC 2007 Question Answering Track. In: TREC. **Anais...** National Institute of Standards and Technology (NIST), 2007. v.Special Publication 500-274.

GLOCKNER, I.; HARTRUMPF, S.; LEVELING, J. **Logical Validation, Answer Merging and Witness Selection A Study in Multi-Stream Question Answering**. 2007.

GRINBERG, D.; LAFFERTY, J.; SLEATOR, D. A Robust Parsing Algorithm for Link Grammars. In: IN PROCEEDINGS OF THE FOURTH INTERNATIONAL WORKSHOP ON PARSING TECHNOLOGIES. **Anais...** [S.l.: s.n.], 1995.

GRUBER, T. R. A translation approach to portable ontology specifications. **Knowl. Acquis.**, London, UK, UK, v.5, p.199–220, June 1993.

HARTRUMPF, S. **Hybrid Disambiguation in Natural Language Analysis**. Osnabrück, Germany: Der Andere Verlag, 2003. ISBN 3-89959-080-5.

HELBIG, H. **Knowledge Representation and the Semantics of Natural Language (Cognitive Technologies)**. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

INCORPORATING NON-LOCAL INFORMATION INTO INFORMATION EXTRACTION SYSTEMS BY GIBBS SAMPLING. **Anais...** [S.l.: s.n.], 2005. p.pp. 363–370.

KAISSER, M. QuALiM at TREC 2005: web-question answering with framenet. In: TREC. **Anais...** National Institute of Standards and Technology (NIST), 2005. v.Special Publication 500-266.

KO, J.; SI, L.; NYBERG, E. A Probabilistic Framework for Answer Selection in Question Answering. In: IN PROCEEDINGS OF NAACL/HLT. **Anais...** [S.l.: s.n.], 2007.

KWOK, C.; ETZIONI, O.; WELD, D. S. Scaling question answering to the web. **ACM Trans. Inf. Syst.**, New York, NY, USA, v.19, p.242–262, July 2001.

LEIDNER, J. L.; CALLISON-BURCH, C. Evaluating Question Answering Systems Using FAQ Answer Injection. In: IN 6TH ANNUAL CLUK RESEARCH COLLOQUIUM. **Anais...** [S.l.: s.n.], 2003.

LIN, D. Dependency-based Evaluation of MINIPAR. In: WORKSHOP ON THE EVALUATION OF PARSING SYSTEMS. **Proceedings...** [S.l.: s.n.], 1998. (Granada).

MAGNINI, B. et al. Comparing Statistical and Content-Based Techniques for Answer Validation on the Web. In: IN PROCEEDINGS OF THE VIII CONVEGNO AI*IA. **Anais...** [S.l.: s.n.], 2002.

MARRAFA, P. **WordNet do Português** : uma base de dados de conhecimento linguístico. [S.l.]: Instituto Camões, 2002.

MILLER, G. A. et al. WordNet: an on-line lexical database. **International Journal of Lexicography**, [S.l.], v.3, p.235–244, 1990.

MOLLA, D.; VICEDO, J. L. Question Answering in Restricted Domains: an overview. **Comput. Linguist.**, Cambridge, MA, USA, v.33, p.41–61, Mar. 2007.

NYBERG, E. et al. The JAVELIN Question-Answering System at TREC 2002. In: TREC 12. **Proceedings...** [S.l.: s.n.], 2002.

OH, H.-J.; MYAENG, S. H.; JANG, M.-G. Effects of answer weight boosting in strategy-driven question answering. **Inf. Process. Manage.**, Tarrytown, NY, USA, v.48, p.83–93, January 2012.

OLIVEIRA, H. G. et al. PAPEL: a dictionary-based lexical ontology for portuguese. In: COMPUTATIONAL PROCESSING OF THE PORTUGUESE LANGUAGE (PROPOR). **Proceedings...** Springer, 2008. p.31–40. (LNAI, v.5190).

PETERS, C. What Happened in CLEF 2009. In: PETERS, C. et al. (Ed.). **Multilingual Information Access Evaluation I. Text Retrieval Experiments**. [S.l.]: Springer Berlin Heidelberg, 2010. p.1–12. (Lecture Notes in Computer Science, v.6241).

PRESTES, K. et al. Extração e Validação de Ontologias a partir de Recursos Digitais. In: IN PROCEEDINGS OF THE 6TH INTERNATIONAL WORKSHOP ON META-MODELS, ONTOLOGIES AND SEMANTIC TECHNOLOGIES. **Anais...** [S.l.: s.n.], 2011. p.183–188.

QI, H. et al. The University of Michigan at TREC 2002: question answering and novelty tracks. In: TREC'02. **Anais...** [S.l.: s.n.], 2002. p.–1–1.

QUARTERONI, S. **Advanced Techniques For Personalized, Interactive Question Answering**. 2007. Tese (Doutorado em Ciência da Computação) — The University of York, York, United Kingdom.

SAIAS, J.; QUARESMA, P. The University of Evora's Participation in QA@CLEF-2007. In: CLEF. **Anais...** Springer, 2007. p.316–323. (Lecture Notes in Computer Science, v.5152).

SALLOUM, W. **A Question Answering System Based on Conceptual Graph Formalism**. 2009.

SILVA, B. C. D. da. A contrução da base da Wordnet.Br: conquistas e desafios. In: SPRINGER. **Anais...** [S.l.: s.n.], 2005.

SRIHARI, R.; LI, W. A question answering system supported by information extraction. In: APPLIED NATURAL LANGUAGE PROCESSING, Stroudsburg, PA, USA. **Proceedings...** Association for Computational Linguistics, 2000. p.166–172. (ANLC '00).

SUZUKI, J.; SASAKI, Y.; MAEDA, E. SVM answer selection for open-domain question answering. In: COMPUTATIONAL LINGUISTICS - VOLUME 1, 19., Stroudsburg, PA, USA. **Proceedings...** Association for Computational Linguistics, 2002. p.1–7. (COLING '02).

VOORHEES, E. M. The TREC-8 Question Answering Track Report. In: IN PROCEEDINGS OF TREC-8. **Anais...** [S.l.: s.n.], 1999. p.77–82.

WILKENS, R. et al. COMUNICA - A Question Answering System for Brazilian Portuguese. In: COLING (DEMOS). **Anais...** Demonstrations Volume, 2010. p.21–24.

ZAMIR, O.; ETZIONI, O. Grouper: a dynamic clustering interface to web search results. In: DEMONSTRATIONS VOLUME. **Anais...** [S.l.: s.n.], 1999. p.1361–1374.

ZHENG, Z. AnswerBus question answering system. In: HUMAN LANGUAGE TECHNOLOGY RESEARCH, San Francisco, CA, USA. **Proceedings...** Morgan Kaufmann Publishers Inc., 2002. p.399–404. (HLT '02).