

**ANDRÉ DE ALMEIDA BARBOSA CECCARELLI**

**GUIA PARA SELEÇÃO DE SISTEMAS DO TIPO PERGUNTA-  
RESPOSTA**

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para conclusão do curso de MBA em Tecnologia de Software.

São Paulo  
2015

**ANDRÉ DE ALMEIDA BARBOSA CECCARELLI**

**GUIA PARA SELEÇÃO DE SISTEMAS DO TIPO PERGUNTA-  
RESPOSTA**

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de MBA em Tecnologia de Software.

Área de Concentração: Tecnologia de Software

Orientador: Prof. Dr. Kechi Hirama

São Paulo  
2015

**ANDRÉ DE ALMEIDA BARBOSA CECCARELLI**

**GUIA PARA SELEÇÃO DE SISTEMAS DO TIPO PERGUNTA-  
RESPOSTA**

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de MBA em Tecnologia de Software.

Área de Concentração: Tecnologia de Software

Orientador: Prof. Dr. Kechi Hirama

São Paulo  
2015

## **DEDICATÓRIA**

*Dedico este trabalho a minha família que sempre me apoia nas minhas decisões e aos meus professores do PECE tão importantes na minha formação acadêmica e profissional.*

## **AGRADECIMENTOS**

À Universidade de São Paulo – USP por abrir espaço para este curso.

À Escola Politécnica da Universidade de São Paulo – EPUSP que forneceu toda a estrutura necessária para a realização deste curso.

Ao PECE – Programa de Educação Continuada em Engenharia que provedu bases de conhecimento essenciais para a elaboração deste trabalho.

Ao meu orientador Prof. Dr. Kechi Hirama pela paciência, suporte e incentivo durante o curso e durante o período de produção deste trabalho.

Aos professores que me guiaram durante o curso.

A minha família que sempre me deu suporte para seguir com as minhas aspirações.

## **RESUMO**

Sistemas do tipo pergunta-resposta (QAS – *Question-Answering System*) têm como objetivo extrair conteúdo conciso e relevante de fontes diversas em resposta a questões feitas em linguagem natural. Este tipo de sistema tem a capacidade de atender a uma das demandas atuais mais presente: extrair de forma natural e personalizada informações da enorme, e também crescente, quantidade de dados existente.

Porém, desenvolver um QAS é de extrema complexidade. Envolve uma variedade de técnicas e mecanismos avançados que devem trabalhar em conjunto para alcançar o efeito desejado. Somado a isto, pode-se considerar que o número de perguntas passíveis de serem compostas é infinito. Estas características acabam abrindo espaço para a constante reinterpretação dos modelos existentes.

Neste cenário, é recomendável haver direcionadores que facilitem interessados no tema a compreender diferentes propostas de QAS. Este trabalho almeja atender exatamente a esta demanda, estruturando um guia prático e dinâmico para avaliar sistemas de QAS e permitir a seleção daquele que mais se alinhe às necessidades do avaliador.

## **ABSTRACT**

Question-Answering System (QAS) is designed to extract concise and relevant content from a variety of sources answering questions asked in natural language. This type of system has the capacity to solve one of the biggest current problems: it extracts information, in a natural and personalized way, from a huge, and also growing, amount of data.

However, it is extremely complex to develop a QAS. It calls a variety of advanced techniques and mechanisms that must work together to reach the desired effect. Besides, the number of questions that can be settled can be considered unlimited. This scenario stimulates the continuous reinterpretation of existing system models.

This situation asks for drivers that can help people to review and choose QAS proposals. The current work aims to find a solution for this request, structuring a practical and dynamic guide that allows evaluating different QAS and selecting the one more aligned to the evaluator's needs.

## LISTA DE ILUSTRAÇÕES

Figura 1. Arquitetura .....	17
Figura 2. Identidade .....	24
Figura 3. Módulos arquiteturais .....	34
Figura 4. Componentes arquiteturais .....	37
Figura 5. Identidade & Arquitetura.....	49
Figura 6. Guia de seleção .....	51
Figura 7. <i>Checklist</i> .....	52
Figura 8. Aplicação – <i>Checklist</i> .....	59

## LISTA DE TABELAS

Tabela 1. Implementações & Pergunta .....	27
Tabela 2. Implementações & Escopo .....	29
Tabela 3. Implementações & Contexto .....	31
Tabela 4. Implementações & Acesso .....	33
Tabela 5. Implementações & Agregação de elementos e propriedades .....	39
Tabela 6. Implementações & Qualificação das variáveis direcionadoras .....	39
Tabela 7. Implementações & Coordenação e adaptação das fontes de dados .....	41
Tabela 8. Implementações & Pesquisa de dados.....	42
Tabela 9. Implementações & Tratamento dos dados coletados .....	44
Tabela 10. Implementações & Constituição da resposta.....	45
Tabela 11. Implementações & Exposição da resposta .....	46
Tabela 12. Implementações & Aprendizado .....	47
Tabela 13. Resumo <i>Checklist</i> .....	55
Tabela 14. Visão Escopo e Diferenciais .....	56
Tabela 15. Visão Implementação .....	57
Tabela 16. Visão Implementação – Seleção de 2 sistemas.....	58
Tabela 17. Aplicação – Visão Escopo e Diferenciais .....	60
Tabela 18. Aplicação – Visão Implementação.....	61
Tabela 19. Cenários de resposta – <i>Checklist</i> .....	63

## **LISTA DE ABREVIATURAS E SIGLAS**

IR	Information Retrieval
NERC	Named Entity Recognition and Classification
NLP	Natural Language Processing
POST	Part-Of-speech Tagging

## SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>12</b>
1.1. Motivações .....	12
1.2. Objetivo .....	14
1.3. Justificativa.....	14
1.4. Estrutura do Trabalho .....	15
<b>2. IMPLEMENTAÇÕES DE QAS.....</b>	<b>16</b>
2.1. Conceitualização .....	16
2.1.1. Definição básica .....	16
2.1.2. Exemplo arquitetural .....	16
2.2. Implementações .....	18
2.3. Considerações do Capítulo.....	22
<b>3. FUNDAMENTOS DE ANÁLISE.....</b>	<b>24</b>
3.1. Identidade.....	24
3.1.1. Pergunta .....	25
3.1.2. Escopo .....	28
3.1.3. Contexto.....	30
3.1.4. Acesso .....	32
3.2. Arquitetura .....	34
3.2.1. Direcionamento da Resposta .....	38
3.2.1.1. Inserção da pergunta.....	38
3.2.1.2. Agregação de elementos e propriedades .....	38
3.2.1.3. Qualificação das variáveis direcionadoras.....	39
3.2.2. Coleta de Informação .....	41
3.2.2.1. Coordenação e adaptação das fontes de dados .....	41
3.2.2.2. Pesquisa de dados .....	42
3.2.2.3. Tratamento dos dados coletados .....	43
3.2.3. Definição da Resposta .....	44
3.2.3.1. Constituição da resposta .....	45
3.2.3.2. Exposição da resposta .....	46
3.3. Aprendizado .....	47
3.4. Considerações do Capítulo.....	48
<b>4. GUIA DE SELEÇÃO .....</b>	<b>51</b>

4.1. Checklist.....	51
4.2. Visão Escopo e Diferenciais .....	56
4.3. Visão Implementação .....	57
4.4. Aplicação .....	58
4.5. Considerações do Capítulo.....	64
<b>5. CONSIDERAÇÕES FINAIS .....</b>	<b>65</b>
5.1. Contribuições do Trabalho.....	66
5.2. Trabalhos Futuros .....	66
<b>REFERÊNCIAS .....</b>	<b>68</b>
<b>BIBLIOGRAFIA COMPLEMENTAR .....</b>	<b>69</b>
<b>APÊNDICE A – Detalhamento das Implementações .....</b>	<b>76</b>
<b>APÊNDICE B – Reuso e Métricas .....</b>	<b>85</b>
<b>APÊNDICE C – Alternativas Validadoras por Implementação .....</b>	<b>89</b>

## 1. INTRODUÇÃO

Este capítulo apresenta motivações, objetivo, justificativas ao tema e estrutura do trabalho.

### 1.1. Motivações

Sistemas do tipo pergunta-resposta (QAS – *Question-Answering System*) têm como objetivo extraír conteúdo conciso e relevante de fontes diversas em resposta a questões feitas em linguagem natural (COMAS, TURMO e MÀRQUEZ, 2012, p. 2). Para isto, os QAS combinam resultados de múltiplos algoritmos buscando um único resultado final, representado pela resposta (LIU e NYBERG, 2013, p. 79). São utilizados, por exemplo, algoritmos relacionados à coleta da informação (IR – *Information Retrieval*), à extração de informação (*Information Extraction*) e ao processamento de linguagem natural (NLP – *Natural Language Processing*) (WANG, 2006, p. 1).

Os QAS têm atraído atenção porque envolvem a ideia de sumarização linguística, capacitando máquinas a gerar respostas concisas e relevantes respeitando valores, regras e relacionamentos linguísticos tipicamente humanos. Eles respondem, assim, a um dos problemas atuais mais desafiadores, que é a crescente discrepância entre o rápido crescimento da quantidade de informação disponível e a praticamente estável capacidade cognitiva e de processamento de informações dos homens (KACPRZYK e ZADROZNY, 2013, p. 33).

A construção de um QAS é complexa devido ao número de variáveis que cercam a sua estruturação. Primeiro, tem-se o ambiente. Este é definido, por exemplo, pela relação dos sistemas com as fontes pesquisadas e pelo tipo de pergunta (fato, comparação, definição, aferição etc.). Em segundo lugar, tem-se a rotina de processamento das informações, sendo esta rotina dividida tradicionalmente em três módulos: 1) classificação da questão, 2) coleta da informação e 3) extração de respostas (WALKE e KARALE, 2013, p. 402 - 404). Por fim, cada um desses módulos utiliza algoritmos diversos. Incluem, por exemplo, análise léxica e morfológica, técnicas de reconhecimento e classificação de entidades (NERC –

*Named Entity Recognizer and Classifier*), técnicas de aprendizado de máquina (*machine learning*), mecanismos de coleta da informação (modelos IR), métodos de ranqueamento de resultados, modelos de identificação de palavras chave, entre tantos outros (COMAS, TURMO e MÀRQUEZ, 2012, p. 6).

Programas e fóruns que se relacionam ao tema foram criados (COMAS, TURMO e MÀRQUEZ, 2012). Três exemplos são:

- *Text Retrieval Conference* (TREC): conferência que tem como corresponsáveis o *National Institute of Standards and Technology* (NIST) e o *U.S. Department of Defense*, sendo que este fórum começou como parte integrante do programa TIPSTER Text.
- *Advanced Question Answering for Intelligence* (AQUAINT): patrocinado também pelo NIST.
- *Conference and Labs of the Evaluation Forum* (CLEF): iniciativa apoiada por integrantes de universidades de países como Itália, França, Espanha, Holanda, Suíça, Suécia e Dinamarca, entre outros.

Estas e outras iniciativas acabaram gerando uma quantidade enorme de estudos e publicações na área, porém sem um ponto de vista único (COMAS, TURMO e MÀRQUEZ, 2012, p. 19:4). Esta dispersão acaba dificultando o entendimento das propostas e sistemas existentes.

A complexidade estrutural deste tipo de sistema somada à diversidade de propostas dificulta a compreensão das capacidades e limites dos QAS. Neste cenário, o estabelecimento de direcionadores que auxiliem interessados no tema a entender, avaliar e selecionar propostas de QAS aderentes às suas necessidades torna-se relevante. Trabalhos recentes como de Liu e Nyberg (2013), Bhaskar, Pal e Bandyopadhyay (2012), Kalyanpur, Patwardhan, Boguraev, Chu-Carroll e Lally (2012), Hakimov, Tunc, Akimaliev e Dogdu (2013) e Comas, Turmo e Màrquez (2012) descrevem soluções implementadas e testadas de QAS. Com isto, permitem avaliar de maneira ampla e atualizada este tipo de sistema, criando bases para a identificação de direcionadores ligados aos sistemas pergunta-resposta.

## 1.2. Objetivo

O objetivo deste trabalho é propor um guia de seleção que auxilie na escolha de um QAS. Este é composto a partir da pesquisa, síntese e análise de propostas específicas de implementação realizadas por autores diversos. O guia é dividido em três partes: 1) *Checklist*, 2) Visão Escopo e Diferenciais e 3) Visão Implementação. O *Checklist* é estruturado levando-se em consideração um conjunto de características que direcionam o arranjo de elementos básicos de processamento que compõem um QAS particular. Assim, a instanciação dessas características define quais QAS são capazes de atender as necessidades assinaladas pelo avaliador. Ao final do *Checklist* é gerado o resultado Visão Escopo e Diferenciais, que apresenta de forma resumida a abrangência temática e os principais diferenciais dos sistemas elegíveis. A partir da Visão Escopo e Diferenciais podem ser acessadas informações específicas sobre a rotina de processamento de cada sistema, estas contidas na Visão Implementação, sendo estas informações organizadas em blocos de processamento de propósito específico.

## 1.3. Justificativa

Wang (2006) compara e analisa sob uma ótica unificada propostas de QAS que respondem a perguntas sobre fatos, apresentando pontos de sucesso e deficiências das soluções existentes. Bhaskar, Pal e Bandyopadhyay (2012) apresentam brevemente trabalhos de outros autores que se relacionam a QAS que respondem a perguntas comparativas e aferitivas. Comas, Turmo e Márquez (2012) summarizam e comparam QAS sob uma perspectiva arquitetural visando analisar como estes trabalham com mecanismos de reconhecimento automático de fala (ASR – *Automatic Speech Recognition*).

Analizando a literatura relacionada à área de QAS, porém, não foi encontrado até o momento um trabalho que compare modelos diferentes de QAS de forma ampla visando criar direcionadores que auxiliem no entendimento de qualquer solução de QAS existente.

Sem um guia prático, a compreensão das variáveis que envolvem a estruturação de um sistema pergunta-resposta torna-se lenta e penosa. O guia aqui proposto é traduzido em um modelo de avaliação e seleção de QAS. Este agiliza o entendimento de QAS específicos e torna mais certeira a escolha de uma solução que se alinhe às necessidades particulares de interessados neste tipo de sistema.

#### 1.4. Estrutura do Trabalho

O Capítulo 1 INTRODUÇÃO apresenta as motivações, o objetivo, as justificativas e a estrutura do trabalho.

O Capítulo 2 IMPLEMENTAÇÕES DE QAS apresenta implementações de QAS estruturadas por autores diversos gerando alicerces para sustentar o Capítulo 3.

O Capítulo 3 FUNDAMENTOS DE ANÁLISE delineia os fundamentos que sustentam a composição do guia de seleção apresentado no Capítulo 4.

O Capítulo 4 GUIA DE SELEÇÃO apresenta o guia de seleção e cenários de utilização do modelo.

O Capítulo 5 CONSIDERAÇÕES FINAIS descreve as conclusões deste trabalho e possibilidades de trabalhos futuros.

REFERÊNCIAS lista o conjunto de fontes utilizadas para a elaboração deste trabalho.

BIBLIOGRAFIA COMPLEMENTAR lista o conjunto de fontes pesquisadas.

APÊNDICE A detalha o processamento dos QAS descritos no Capítulo 2.

APÊNDICE B lista modelos e fontes de terceiros utilizados pelos autores das implementações expostas no Capítulo 2, além de métricas utilizadas na avaliação das implementações.

APÊNDICE C mapeia a validação de cada uma das implementações apresentadas no Capítulo 2 por item do *Checklist*.

## 2. IMPLEMENTAÇÕES DE QAS

A complexidade estrutural dos QAS dificulta o entendimento de sistemas existentes. Este capítulo apresenta algumas propostas discutidas na literatura para entender melhor os conceitos envolvidos.

Na Seção 2.1 são apresentados, de forma breve, conceitos relacionados aos QAS visando auxiliar na leitura da Seção 2.2. A Seção 2.2 traz cinco propostas de implementação de QAS desenvolvidas por autores diversos, abordando rotinas de processamento do ciclo completo de pergunta-resposta.

### 2.1. Conceitualização

Nesta seção é apresentada uma definição básica posicionando qual é o objetivo principal de qualquer QAS. Também é exposta uma arquitetura de alto nível visando oferecer uma estrutura que apoie o entendimento de cada uma das propostas de implementação apresentadas neste capítulo.

#### 2.1.1. Definição básica

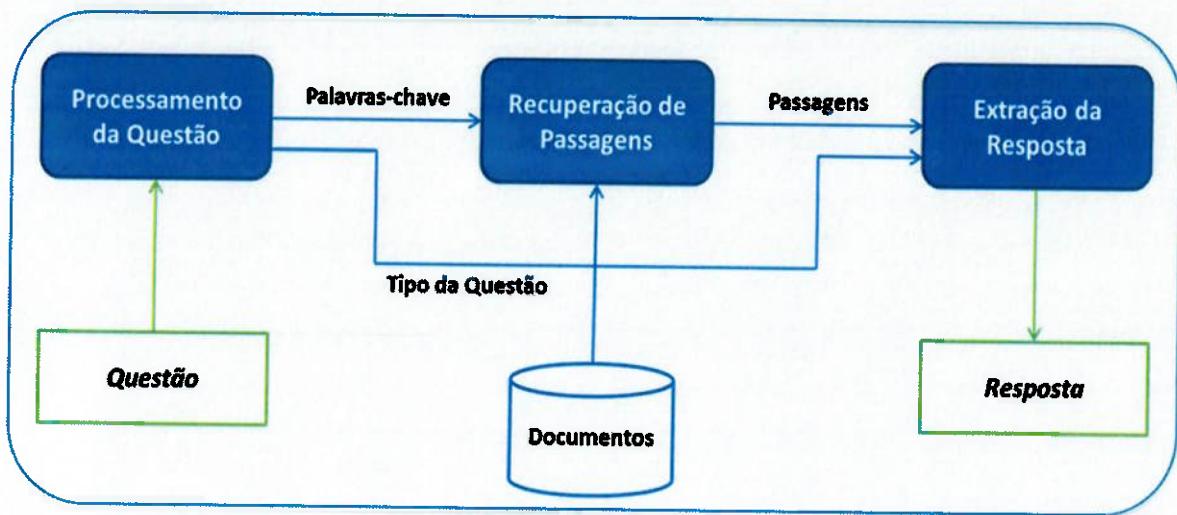
Sistemas do tipo pergunta-resposta (QAS – *Question-Answering System*) buscam extrair conteúdo conciso e relevante de uma ou de diversas fontes de informação, respondendo a questões feitas em linguagem natural (COMAS, TURMO e MÀRQUEZ, 2012, p. 2). Atendem às demandas de um usuário respeitando valores, regras e relacionamentos linguísticos tipicamente humanos (KACPRZYK e ZADROZNY, 2013, p. 33). Para isto, comumente, utilizam regras heurísticas, estatísticas e linguísticas.

#### 2.1.2. Exemplo arquitetural

Comas, Turmo e Màrquez (2012, p.4-5) descreveram uma arquitetura de alto nível para sistemas que tratam de responder a questões sobre fatos. Um exemplo de questão sobre fato é: Qual é o ponto de ebulação da água ao nível do mar?

A arquitetura é apresentada na Figura 1. Nesta encontram-se módulos e as informações trocadas entre eles.

**Figura 1. Arquitetura**



Fonte: Adaptado de Comas, Turmo e Márquez (2012, p. 4-5)

O processamento nesta arquitetura se divide em três módulos sequenciais:

1. *Processamento da Questão*: processa a pergunta para extrair termos de pesquisa e determina qual tipo de resposta é esperado.
2. *Recuperação de Passagens*: abalizado pelas informações do módulo anterior, realiza a função de recuperação de informações (IR – *Information Retrieval*) buscando por documentos ou trechos que possam conter a resposta.
3. *Extração da Resposta*: classifica as informações recuperadas e cruza estas classificações com as classificações obtidas no Processamento da Questão para depois utilizar métodos de ranqueamento ordenando as respostas candidatas com vista a definir a resposta final.

Os autores ressaltam que esta é uma arquitetura genérica e que sistemas específicos podem apresentar variações. Por exemplo, podem ser encontradas diferenças nas informações utilizadas por cada módulo e como estas fluem. Mecanismos de retorno e realimentação entre os módulos e de pré-processamento de documentos também podem ser implementados.

Esta arquitetura, por ser de alto nível, é um bom guia para auxiliar no entendimento das implementações descritas na próxima seção, independentemente das variações dos modelos apresentados.

## 2.2. Implementações

Esta seção descreve de forma resumida cinco diferentes propostas de implementação de QAS estruturadas por autores diversos.

As propostas foram selecionadas dentro de um conjunto de mais de 60 artigos encontrados principalmente nas bases da IEEE Xplore Digital Library<sup>1</sup> e ACM Digital Library<sup>2</sup>. A seleção levou em consideração, principalmente, duas características: 1) atualidade e 2) descrição do processamento desde o tratamento da pergunta até a definição da resposta.

Maiores detalhes são apresentados no Apêndice A e B. O Apêndice A acrescenta informações concernentes ao modelo de processamento. O Apêndice B mapeia situações de reuso e métricas.

## IMPLEMENTAÇÃO 1

Liu e Nyberg (2013), em seu artigo *A phased ranking model for question answering*, trabalharam sobre uma proposta que estimula a propagação e inter-relação de resultados obtidos por diferentes rotinas de processamento. Seu modelo de ranqueamento é apoiado por um gráfico sistêmico de dependência de objetos. Este gráfico mapeia os objetos criados, relacionando suas características e dependências. Os autores integraram sua proposta ao OpenEphyra, um *framework*

---

<sup>1</sup> <http://ieeexplore.ieee.org/Xplore/home.jsp>

<sup>2</sup> <http://dl.acm.org/>

modular e extensível desenvolvido dentro do projeto Ephyra<sup>3</sup>. O sistema trabalha com perguntas relacionadas a fatos, como: Em qual continente fica o Brasil?

No modelo proposto, a interpretação da pergunta identifica palavras-chave que são utilizadas para compor *queries* que coletam informações relevantes de múltiplas fontes de dados. Os trechos coletados são analisados e comparados com o tipo de resposta esperado. Por fim, os trechos são ranqueados, sendo que a resposta posicionada como primeira é a dada como resposta final. Para o ranqueamento são utilizados os atributos gerados em cada ponto do gráfico sistêmico de objetos (entrada, dependência e saída) combinando-os linearmente com o objetivo de gerar uma pontuação normalizada.

## IMPLEMENTAÇÃO 2

Bhaskar, Pal e Bandyopadhyay (2012), em seu artigo *Comparative & Evaluative QA System In Tourism Domain*, trabalharam um sistema que responde especificamente a perguntas relacionadas à área de Turismo e que exigem comparações e aferições. Para isto, definiram mecanismos que reconhecem expressões relacionadas a comparações e aferições, que identificam as necessidades do usuário e que predizem características não claramente definidas. Trabalharam na transformação de expressões qualitativas em critérios objetivos de avaliação, quando este se faz necessário.

No modelo proposto, a interpretação da pergunta mapeia o que o usuário deseja obter como resposta. Muito deste trabalho é apoiado por uma base de conhecimento específico que permite ao sistema entender como um ser humano interpretaria as informações apresentadas. Interpretada a pergunta, é iniciada a busca por dados que possam conter a resposta desejada. O ponto turístico, que pode ser específico (ex. Monte Everest) ou genérico (ex. Montanha), é fundamental para direcionar a busca. São utilizadas fontes de dados que apresentam padrões de organização e de

---

<sup>3</sup> <http://www.ephyra.info/>

classificação de informação bem delineados. Os trechos coletados são analisados para identificação de atributos, que são valorados conforme regras pré-estabelecidas na base de conhecimento específico do domínio. As pontuações obtidas são somadas de forma ponderada, sendo as cinco maiores consideradas como resposta final.

### IMPLEMENTAÇÃO 3

Kalyanpur, Patwardhan, Boguraev, Chu-Carroll e Lally (2012), em seu artigo *Parallel And Nested Decomposition For Factoid Questions*, propuseram um *framework* para trabalhar perguntas que relacionam dois ou mais fatos, como: Quem é o homem do qual a filha, para honrar seu trabalho, adotara o nome de Maria Celeste quando se tornou freira em 1616? Integraram seu *framework* ao QAS Watson, da IBM. Para atender ao seu propósito, os autores estruturaram mecanismos para decompor os fatos e estabeleceram meios para selecionar a melhor resposta conforme a relação existente entre os fatos.

Durante a interpretação da pergunta, o sistema trabalha para reconhecer e tipificar os múltiplos fatos da questão, gerando subquestões. Também é definido o tipo de dependência entre as subquestões: Paralelo ou Aninhado.

Na dependência Paralela as subquestões são tratadas de forma independente. É processada individualmente cada uma das subquestões e gerada uma lista ranqueada de trechos advindos de fontes de dados diversas. Os trechos coletados são sintetizados e re-rankeados. As listas geradas pelas diversas subquestões são unificadas. Correndo paralelamente ao processamento das subquestões, também é processada a pergunta sem decompô-la, encontrando uma lista de trechos para esta. Esta lista é cruzada com a lista gerada pelo tratamento das subquestões, visando reavaliar o ranqueamento dos trechos, sendo que o trecho que ficar mais bem posicionado é selecionado como resposta final.

Na dependência Aninhada as subquestões são processadas conforme ordem lógica determinada por qual subquestão “contém” a outra. Primeiro são coletados trechos para a subquestão “contida”, sendo estes ranqueados e utilizados para reescrever a

subquestão “contém” compõndo nova subquestão. A subquestão reescrita é utilizada para uma nova coleta de informações. Correndo paralelamente ao processamento das subquestões, também é processada a pergunta sem decompô-la, encontrando uma lista de trechos para esta. A resposta final é selecionada pelo cruzamento da lista obtida por meio das subquestões com a lista gerada pelo tratamento da pergunta não decomposta.

#### IMPLEMENTAÇÃO 4

Hakimov, Tunc, Akimaliev e Dogdu (2013), em seu artigo *Semantic QAS Over Linked Data Using Relational Patterns*, modelaram um QAS que utiliza o *Resource Description Framework* (RDF), um modelo padrão de intercâmbio de dados da Web. O sistema está amarrado ao DBpedia, que é um banco de dados Web com informações estruturadas extraídas da Wikipedia. Trabalharam com perguntas relacionadas a fatos.

No modelo proposto, interpreta-se a pergunta com o objetivo de traduzi-la para um padrão RDF de três elementos alinhado à base ontológica da DBpedia. Para isto, a pergunta é decomposta formando uma árvore de dependência, que é analisada recursivamente. Durante a interpretação da pergunta também é definido o tipo de resposta esperado. Com base nas informações obtidas são estruturadas *queries* que coletam informações da DBpedia. Para cada conjunto de três elementos trabalhado pelas *queries* é atrelada uma pontuação. O sistema checa o alinhamento entre as informações coletadas e o tipo de resposta esperado. Os candidatos que estão alinhados são ranqueados e o candidato que aparece na primeira posição é considerado como resposta final.

#### IMPLEMENTAÇÃO 5

Comas, Turmo e Márquez (2012), em seu artigo *Sibyl, A Factoid QAS For Spoken Documents*, apresentaram um QAS otimizado para trabalhar com fontes de dados de base oral. Aperfeiçoaram técnicas de recuperação de informação utilizando conceitos como *Dynamic Query Relaxation Procedure* e trabalharam com diferentes camadas de informação linguística, que incluem identificação de entidades por meio

de informações fonéticas, análise sintática aplicada a transcrições de falas e correlacionamento de termos. Focaram seu modelo para responder a perguntas relacionadas a fatos.

No modelo proposto, a interpretação da pergunta identifica o tipo esperado de resposta e um conjunto de palavras-chave, sendo que são atreladas às palavras-chave pontuações de relevância conforme modelo heurístico pré-definido. Dois modelos são utilizados para a coleta de dados. O primeiro modelo processa o conjunto de palavras-chave estruturando *queries* dinâmicas e utiliza a ferramenta Lucene IR (Apache). O segundo modelo utiliza similaridade fonética para encontrar as palavras-chave, melhorando a interpretação dos trechos de documentos obtidos por meio de mecanismos de reconhecimento de fala. Os trechos que atendem as características identificadas no processo de interpretação da pergunta são ranqueados, sendo a resposta final o que aparece na primeira posição.

### 2.3. Considerações do Capítulo

Percebe-se pela análise das implementações que ocorrem reinterpretações de mecanismos diversos, como processamento de linguagem natural, estruturação de *queries* de pesquisa, recuperação de informação etc. em busca de um molde que consiga atender às demandas que giram no entorno de um sistema do tipo pergunta-resposta. Esta situação pode ser relacionada a uma série de fatores, como complexidade dos mecanismos de processamento, variação dos modelos de interpretação conforme foco da pergunta e disponibilidade e qualidade dos dados.

Mesmo as estruturas mais complexas de QAS acabam atendendo apenas a uma fração das perguntas passíveis de serem formuladas por seres humanos. Este fato fica evidente quando é observado que as implementações da Seção 2.2 acabam se voltando a uma capacidade específica de geração de resposta. E mesmo limitando seu escopo, ao serem analisados os resultados apresentados dentro dos artigos, percebe-se uma margem significativa de perguntas não respondidas ou respondidas de forma incorreta.

Este cenário evidencia as dificuldades que acompanham o desenvolvimento de um QAS. Porém, apesar das dificuldades, o contínuo desenvolvimento deste tipo de sistema oferece benefícios potenciais enormes para todos os setores da sociedade. Estes benefícios podem estar ligados a uma melhora nas tomadas de decisão de negócio, podem passar pela criação de novos modelos de ensino ou mesmo na obtenção de melhorias na área médica, como no processo de diagnóstico de doenças. Assim, as dificuldades existentes não devem impedir avanços nesta área.

Ter uma forma de analisar os sistemas hoje existentes e ter a capacidade de utilizar o mesmo modelo para analisar QAS que ainda podem ser criados se torna uma demanda importante. No próximo capítulo, são estruturados os alicerces para construção de um modelo de avaliação que ajude a entender e comparar diferentes propostas de QAS, derivando deste um guia de seleção a ser exposto no Capítulo 4.

### 3. FUNDAMENTOS DE ANÁLISE

O Capítulo 3 apresenta fundamentos de análise que auxiliam a traduzir propriedades de diferentes QAS sob perspectivas comuns. A Seção 3.1 apresenta a Identidade que define as capacidades e limites de um QAS, a Seção 3.2 apresenta a Arquitetura que ajuda a comparar diferentes modelos de implementação e a Seção 3.3 tece um breve comentário sobre a relação entre as técnicas de aprendizado (*machine learning*) e QAS.

#### 3.1. Identidade

A Identidade define características de alta importância para a acepção de um QAS. Estas características influenciam diretamente na estratégia de implementação a ser adotada e definem as capacidades e limites que o sistema terá. Um potencial usuário, por meio da Identidade, consegue verificar se o sistema tem as propriedades necessárias para atender as suas necessidades.

A Figura 2 mostra que a Identidade é composta por quatro características: Pergunta, Escopo, Contexto e Acesso.

**Figura 2. Identidade**



Fonte: o autor

As características apresentadas são detalhadas nas próximas subseções.

### 3.1.1. Pergunta

A característica Pergunta representa a capacidade do sistema de entender perguntas e gerar respostas conforme este entendimento. Está relacionada à estrutura interpretativa, que pode ser mais ou menos complexa conforme o nível de subjetividade que visa atender.

O nível de subjetividade espelha o número de maneiras com as quais uma informação pode ser observada. Por exemplo, perguntas sobre fatos pontuais, como a data de nascimento de uma pessoa, apresentam baixo nível de subjetividade. O nível de interpretação exigido para perguntas como esta é baixa, já que tendem a conter apenas uma resposta. Assim, são mais fáceis de serem interpretadas e respondidas. Já perguntas que pedem para definir algo como “melhor” exigem a conceitualização de “melhor” e este pode apresentar variações. Assim, há um aumento do nível de subjetividade e, consequentemente, aumenta o nível de exigência interpretativa.

A complexidade da característica Pergunta pode ser atrelada ao número de conversões e de conferências de informação necessários para se chegar a melhor resposta. Por exemplo, uma pergunta sobre um fato precisa identificar qual é o fato a ser respondido e obter a informação que responde a este fato. Já em uma pergunta sobre dois ou mais fatos, adiciona-se a necessidade de identificar a relação entre os diversos fatos. E no caso de perguntas comparativas é preciso identificar o que é comparado, qual é o conceito de comparação e unir as informações coletadas de forma a representar a comparação desejada.

Os artigos de Wang (2006) e Comas, Turmo e Márquez (2012) apresentaram alguns tipos de Pergunta, sendo possível derivar os seguintes:

I. *Fato*: a pergunta está relacionada à informação bem definida e delimitada sobre um objeto específico. Exemplos:

- Qual é o nome completo da Xuxa?

- Em qual estado está localizado Belo Horizonte?
- Em qual data foi derrubado o muro de Berlim?

II. *Fato Composto*: a pergunta está relacionada à informação bem definida e delimitada sobre dois ou mais objetos específicos. Exemplos:

- Qual era o partido político da base governista que tinha mais cadeiras no senado em 2014?
- Quando foi lançada a sétima edição brasileira do livro Engenharia de Software de Roger S. Pressman?

III. *Comparação*: a pergunta pede comparação entre diferentes objetos seguindo um critério bem definido e delimitado. Exemplos:

- Quais são os hotéis mais baratos da cidade de São Paulo?
- Quem é mais alto, João ou Pedro?

IV. *Aferição*: a pergunta exige a conceitualização de critério quantitativo para responder a um objeto bem definido e delimitado, sendo que a resposta não pode ser encontrada como fato. Exemplos:

- Qual era a minha idade em 1999?
- Quanto cresci de 1999 até 2013?

V. *Definição*: a pergunta pede informações relevantes relacionadas a tópico ou objeto bem definido e delimitado. Exemplos:

- Qual o papel de um juiz?
- O que é economia?

Os tipos de Pergunta definidos acima aparecem de forma frequente na literatura. Entretanto o conjunto apresentado não é exaustivo. Pode-se verificar isto por meio da pergunta: Qual é o papel da religião dentro da sociedade? Este tipo de pergunta se conecta a um ponto de vista, o que permite gerar diferentes respostas e cada uma pode ser considerada tanto correta como incorreta dependendo do perfil do usuário.

Na Tabela 1 são identificados os tipos de Pergunta para cada implementação apresentada na Seção 2.2.

**Tabela 1. Implementações & Pergunta**

Sistema	Pergunta
Implementação 1	Fato
Implementação 2	Comparação; Aferição
Implementação 3	Fato; Fato Composto
Implementação 4	Fato
Implementação 5	Fato

**Fonte:** o autor

Bhaskar, Pal e Bandyopadhyay (2012) fizeram alusão à expressão tipo de pergunta para definir que seu sistema responde a perguntas comparativas e aferitivas (p.458). Hakimov, Tunc, Akimaliev e Dogdu (2013) utilizaram a expressão tipo de pergunta para representar classificações obtidas durante a interpretação da pergunta que são utilizadas para a extração de respostas (p.84). Liu e Nyberg (2013) utilizaram a expressão tipo de pergunta também para se referir a classificações utilizadas para a extração da resposta (p.83). Kalyanpur, Patwardhan, Boguraev, Chu-Carroll e Lally (2012) utilizaram a expressão tipo de pergunta para se referenciar a forma de dependência das subquestões (p. 856).

Assim, o termo Pergunta se alinha a um termo frequentemente utilizado pelos autores citados, que é tipo de pergunta. Porém, os autores citam tipo de pergunta como um termo agrupador de classificações, não o conceitualizando, e percebem-se algumas nuances específicas do termo em cada um dos artigos estudados. Por isso, a definição do termo Pergunta contida nesta subseção é entendimento construído pelo atual trabalho.

### 3.1.2. Escopo

A característica Escopo representa os assuntos e termos passíveis de serem reconhecidos e tratados dada uma estrutura particular de um QAS. A instanciação do Escopo se dá no formato: Restrição (Assunto,..., Assunto). O termo Restrição indica que o sistema é direcionado para uma área específica. O termo Assunto representa vocábulos que sintetizam um grupo de elementos que formam um objeto genérico compreensível aos seres humanos. Segue exemplo de Escopo: Alimentação (Nome, Moeda, Organização). Caso não se delimite fronteiras para os assuntos, o espaço Restrição fica vazio. Segue exemplo: (Nome, Moeda, Organização).

Há uma infinidade de vocábulos que podem ser utilizados para a instanciação do Escopo. A estratégia mais eficiente para mapeamento dos vocábulos é a comparação e análise de termos utilizados dentro de um conjunto pré-definido de sistemas a serem comparados, sendo estabelecida uma linha de raciocínio que gira em torno deste conjunto. Deve-se tomar a precaução de evitar uma multiplicação excessiva de termos, o que acabaria dificultando a comparação dos sistemas. É preferível escolher termos mais abrangentes em detrimento de uma inclusão excessiva de termos específicos. Os conceitos Entidade e Domínio, utilizados pela maioria das implementações apresentadas durante o Capítulo 2, podem auxiliar na definição dos melhores vocábulos.

O conceito Entidade é normalmente identificado na literatura pela nomenclatura *Named Entity* (NE). A base deste conceito leva em consideração que as respostas devem conter elementos aderentes às características da pergunta realizada (WANG, 2006). Segundo Michalski (1983) *apud* Kacprzyk e Zadrożny (2013), entidades são descrições simbólicas semântica e estruturalmente compreensíveis a um ser humano, sendo que os seus elementos devem ser vistos como um único item de informação. Exemplos de entidades encontradas na literatura são Data, Local, Medida, Número, Organização, Pessoa, Hora. As expressões *Named Entity Recognition* (NER) e *Named Entity Recognition and Classification* (NERC), que aparecem com frequência na literatura, englobam as rotinas de identificação de NE (COMAS, TURMO e MÀRQUEZ, 2012).

Em relação ao conceito Domínio, encontram-se dois tipos na literatura: Domínio Aberto e Domínio Especializado. QAS baseados em Domínio Aberto apresentam baixa dependência de terminologias e conceitos específicos relacionados a uma área do conhecimento. Quando um QAS é de Domínio Especializado (Fechado / Restrito), indica que os mecanismos de processamento do QAS apresentam forte relação com terminologias e conceitos específicos de uma área do conhecimento (WALKE e KARALE, 2013, p. 402). Enquanto um QAS de Domínio Aberto responde a um número maior de áreas do conhecimento, um QAS de Domínio Especializado acaba possibilitando responder a perguntas mais complexas.

A Tabela 2 identifica o Escopo para cada implementação apresentada na Seção 2.2.

**Tabela 2. Implementações & Escopo**

Sistema	Escopo
Implementação 1	(Data, Local, Pessoa, Organização)
Implementação 2	Turismo (Itinerário, Acomodação, Transporte, Passeios, Período, Custo)
Implementação 3	(Data, Hora, Local, Endereço, Pessoa, Contatos Pessoais, Endereços Web, Organização, Número, Animal, Doença, Produto, Transporte, Clima, Acontecimento)
Implementação 4	(Data, Local, Número, Organização, Pessoa)
Implementação 5	(Data, Hora, Local, Organização, Pessoa, Número, Medida)

Fonte: o autor

O termo Escopo e sua respectiva definição são especificados pelo atual trabalho. Os conceitos mais próximos de Escopo seriam Entidade e Domínio, porém há motivos para não utilizar estes conceitos como substitutos diretos de Escopo. Primeiro, não há obrigatoriedade em se utilizar os termos relacionados a Entidades e Domínio, quando estes existirem, para instanciar o Escopo. Isto depende da lógica criada em torno do conjunto de sistemas avaliados. É o caso de Bhaskar, Pal e Bandyopadhyay (2012), no qual os termos relacionados ao Contexto da Questão (p.460) apresentam os melhores vocábulos para instanciar o Escopo. Soma-se a isto o caso de que, quando o Domínio é Aberto, não há termos existentes.

Segundo, podem ocorrer adaptações nos termos que representam a Entidade e o Domínio de um sistema específico. A definição dos vocábulos a serem utilizados em Escopo exige um alinhamento entre diversos QAS, enquanto que cada QAS acaba apresentando uma lógica diferente para definir os termos relacionados a Domínio e Entidade. Por exemplo, Liu e Nyberg (2013) utilizaram o conceito de Entidade dentro de uma classificação de três camadas, formada por 7 tipos genéricos de resposta, 44 categorias e 110 subcategorias. Já Bhaskar, Pal, e Bandyopadhyay (2012) utilizaram 11 objetos, representando as entidades, com o objetivo de definir o que está sendo comparado ou avaliado, sendo estes analisados em conjunto com 6 contextos da questão e 10 restrições.

O Escopo abre espaço para trabalhar de maneira mais flexível os vocábulos que o instanciam, permitindo, por exemplo, unificar expressões. Isto é importante já que se visa gerar um guia de seleção que tem como objetivo comparar todo e qualquer sistema de QAS.

### **3.1.3. Contexto**

A característica Contexto representa a capacidade de um QAS de entender o meio e/ou o momento no qual uma pergunta é realizada visando melhorar a interpretação da pergunta e/ou a geração da resposta. Um QAS pode utilizar informações não derivadas diretamente da pergunta, mas sim derivadas do cruzamento da pergunta com um encadeamento de informações inerentes ao meio e ao momento no qual a pergunta está sendo realizada. É esta capacidade que instancia o Contexto.

Algumas perguntas, para serem respondidas, podem depender de complementação de ideia ou termo relacionado especificamente a sessão à qual está sendo realizada. Seguem exemplos abaixo:

- João joga bola?
- Qual é o melhor hotel?

Na primeira pergunta deve ser definido quem é João: Filho? Irmão? Amigo? Já no caso da segunda pergunta a definição da abrangência geográfica pode alterar todo o sentido da resposta: Do Mundo? Do Brasil? De São Paulo?

A Tabela 3 mostra que esta característica é ignorada por quase todos os artigos citados na Seção 2.2. Somente Bhaskar, Pal e Bandyopadhyay (2012) implementam mecanismos que se relacionam ao Contexto. Porém, esta característica tem um enorme potencial para melhorar um QAS. Tem a capacidade, por exemplo, de encurtar perguntas e ao mesmo tempo melhorar a interpretação destas, de personalizar um QAS ao utilizar informações pessoais do usuário, de permitir construir uma dinâmica de diálogo entre homem e máquina.

**Tabela 3. Implementações & Contexto**

Sistema	Contexto
Implementação 1	Não foram especificados mecanismos relacionados ao contexto.
Implementação 2	Contexto instanciado pela criação de modelo de predição de parâmetros de avaliação, interpretando como uma expressão comparativa ou aferitiva deve se comportar para uma pergunta específica.
Implementação 3	Não foram especificados mecanismos relacionados ao contexto.
Implementação 4	Não foram especificados mecanismos relacionados ao contexto.
Implementação 5	Não foram especificados mecanismos relacionados ao contexto.

Fonte: o autor

O termo Contexto aparece dentro da literatura analisada conforme se segue:

- Bhaskar, Pal e Bandyopadhyay (2012) se referenciaram ao termo Contexto para indicar um mecanismo que auxilia na interpretação de expressões comparativas e aferitivas entendendo como estas se encaixam nas necessidades e demandas do usuário (p.459-460).
- Comas, Turmo e Márquez (2012), apesar de não utilizarem nenhum mecanismo do tipo na sua proposta de QAS, comentaram sobre Contexto indicando que há QAS que são capazes de responder a uma série de

questões sequenciais relacionadas a um tópico ou capazes de entrar em um diálogo com um ser humano (p.2).

Analizando de forma conjunta as descrições acima e cruzando estas com o processamento típico de um QAS, o atual trabalho estabeleceu a definição de Contexto descrita nesta subseção.

### **3.1.4. Acesso**

A característica Acesso representa o relacionamento entre os mecanismos de processamento e as fontes de dados consultadas para um QAS particular. Pode ser classificado como Flexível e Rígido.

No Acesso Flexível, a amarração a elementos formais e específicos de associação e de classificação inerentes às fontes de dados consultadas é fraca. Assim, os QAS que utilizam este tipo de acesso conseguem abranger de forma relativamente simples um grande número de fontes. Entretanto há o custo do maior esforço para selecionar a resposta correta, pois para flexibilizar as fontes consultadas evita-se criar tratamentos e seleções prévias. Problemas de informações não confiáveis, repetição de informação, dados conflituosos, entre outros, são efeitos indesejados que tentam ganhar escala neste modelo.

No Acesso Rígido, a amarração a elementos formais e específicos de associação e de classificação inerentes às fontes de dados consultadas é forte. A modelagem dos mecanismos de processamento do QAS, neste cenário, sofre forte interferência de características específicas de um conjunto de fontes de dados. Neste cenário é possível terceirizar parte do esforço empregado pelo QAS ao aproveitar pré-processamentos de dados feitos pelas fontes consultadas. Entretanto há um maior esforço para acrescentar novas fontes e há maiores restrições em relação à variedade de fontes que podem ser utilizadas, o que ocasiona em uma perda de escala. Aqui somente é viável utilizar fontes de dados do tipo semi-estruturadas ou estruturadas.

A Tabela 4 indica o tipo de Acesso de cada implementação apresentada durante a Seção 2.2.

**Tabela 4. Implementações & Acesso**

Sistema	Acesso
Implementação 1	Flexível
Implementação 2	Rígido
Implementação 3	Flexível
Implementação 4	Rígido
Implementação 5	Flexível

Fonte: o autor

O conceito de Acesso apresentado nesta subseção é derivado da análise das propostas de implementação apresentadas pelos trabalhos de Liu e Nyberg (2013), Bhaskar, Pal e Bandyopadhyay (2012), Kalyanpur, Patwardhan, Boguraev, Chu-Carroll e Lally (2012), Hakimov, Tunc, Akimaliev e Dogdu (2013) e Comas, Turmo e Márquez (2012). Liu e Nyberg (2013) criaram um sistema em que a busca por informações ocorre por meio da construção de *queries* que se relacionam a agentes de busca variados que consultam fontes de dados não estruturados. Bhaskar, Pal e Bandyopadhyay (2012) amarraram seu sistema a uma série de tabelas locais que auxiliam na coleta de informações de fontes específicas que tem como características serem semi-estruturadas. Kalyanpur, Patwardhan, Boguraev, Chu-Carroll e Lally (2012) utilizaram a estrutura de busca do QAS Watson, da IBM, para fazer a relação com as fontes de dados, sendo que este QAS visa abranger o maior número possível de fontes para responder a uma grande variedade de perguntas sobre fatos diversos. Hakimov, Tunc, Akimaliev e Dogdu (2013) construíram seu sistema baseado em conceitos específicos como SPARQL queries e RDF de três elementos, coletando informações de uma base de dados específica, a DBpedia. Comas, Turmo e Márquez (2012) trabalharam com dois modelos de coleta de informação. Utilizaram *queries* dinâmicas e mecanismos adaptados para buscar informações em textos derivados de transcrições de falas.

### 3.2. Arquitetura

Por definição, um QAS extrai conteúdo conciso e relevante de uma ou de diversas fontes de informação, respondendo a questões feitas em linguagem natural. Assim, interpretar a pergunta, buscar dados e determinar a resposta são funções básicas de qualquer sistema do tipo pergunta-resposta. Por isto, é possível generalizar uma arquitetura de QAS.

A arquitetura genérica é representada por três módulos de processamento, que expressam rotinas de processamento fundamentais:

1. *Direcionamento da Resposta*: processa a pergunta com o objetivo de mapear, agregar e qualificar informações relevantes que direcione os próximos módulos de processamento.
2. *Coleta de Informação*: estabelece vínculo com as fontes de dados objetivando encontrar, recuperar e tratar informações relevantes para a composição da resposta desejada, respeitando o direcionamento dado pelo módulo anterior.
3. *Definição da Resposta*: respeitando o direcionamento dado no primeiro módulo e com base nas informações obtidas dentro do segundo módulo, são constituídas, validadas e expostas as potenciais respostas.

**Figura 3. Módulos arquiteturais**



**Fonte:** o autor

A Figura 3 mostra o fluxo de processamento entre os módulos. O fluxo natural é Direcionamento da Resposta, Coleta de Informação e Definição da Resposta, indicado pela tonalidade azul escuro nas setas. Porém, pode haver troca de informações entre os módulos que fuja do fluxo natural, como é observado no tratamento de perguntas aninhadas exposta pela Implantação 3 encontrada na Seção 2.2. Este tipo de troca é ressaltado pelo azul claro das setas.

A arquitetura genérica apresentada foi gerada com base nos trabalhos de Comas, Turmo e Márquez (2012) e Walke e Karale (2013), que se somaram à análise das implementações apresentadas na Seção 2.2.

Comas, Turmo e Márquez (2012) definiram a estrutura arquitetural básica de um QAS como:

1. Processamento da Questão (*Question Processing*): processa a pergunta para extrair termos de pesquisa e determina qual tipo de resposta é esperado.
2. Recuperação de Passagens (*Passage Retrieval*): recupera documentos ou trechos utilizando as informações geradas no módulo anterior.
3. Extração da Resposta (*Answer Extraction*): identifica respostas candidatas com base nas informações geradas no módulo Processamento da Questão e ranqueia os potenciais candidatos a resposta.

Walke e Karale (2013) definiram a estrutura arquitetural básica de um QAS como:

1. Módulo de Classificação da Questão (*Question Classification Module*): categoriza as questões permitindo guiar o que o sistema deve procurar e que tipo de resposta este deve gerar.
2. Módulo de Recuperação de Informação (*Information Retrieval Module*): acessa as fontes de dados, verifica estas e recupera informações que tenham potencial para gerar respostas para a pergunta realizada.
3. Módulo de Extração da Informação (*Answer Extraction Module*): identifica, extrai e valida respostas trabalhando os trechos coletados no módulo anterior.

O principal motivo para realizar adaptações nos conceitos apresentados por Comas, Turmo e Márquez (2012) e Walke e Karale (2013) foi a necessidade dos módulos

arquiteturais representarem todo e qualquer tipo de QAS, já que o objetivo do presente estudo é o de construir direcionadores de avaliação para todo e qualquer tipo de QAS apresentado. Por exemplo, no primeiro módulo da arquitetura nenhum dos autores deixa clara a ação agregar informações e no terceiro módulo não abarcam as ações constituir e expor, todas consideradas importantes para atender devidamente aos objetivos do atual trabalho. Os nomes dos módulos foram adaptados conforme motivos apresentados abaixo:

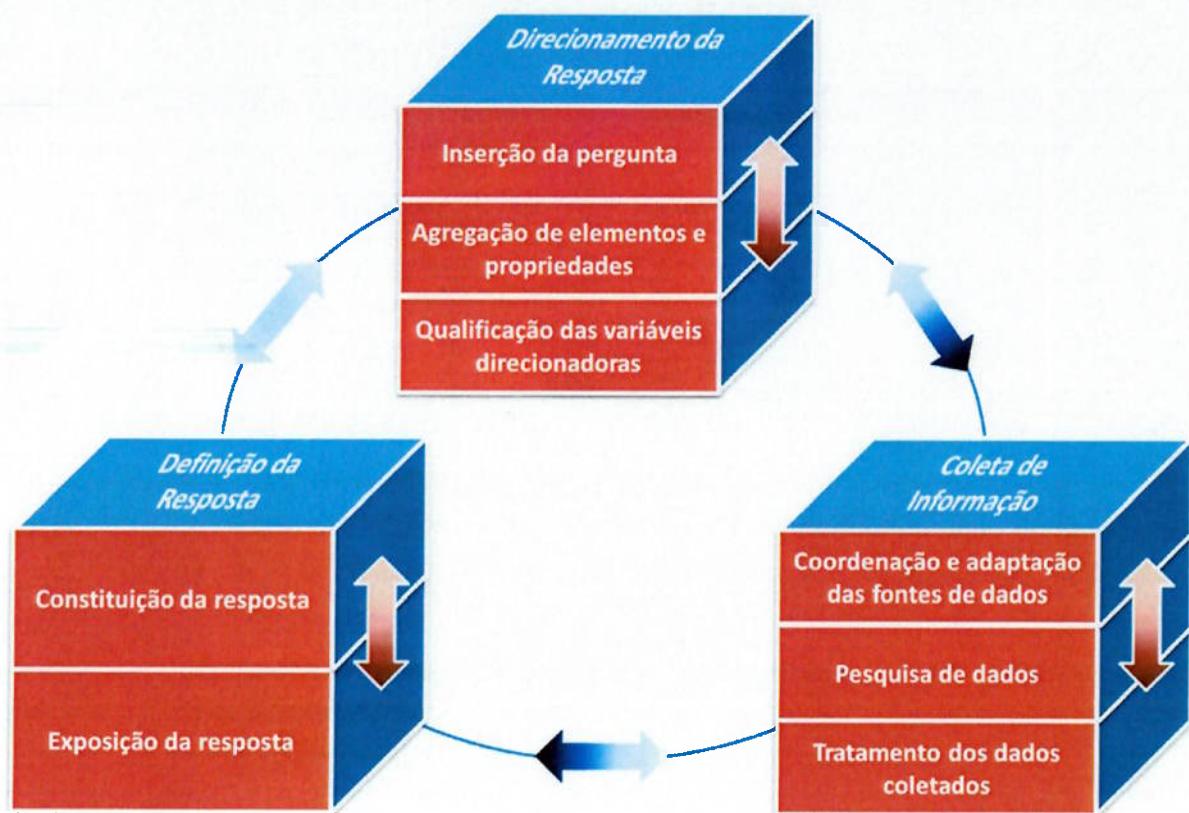
- Direcionamento da Resposta: ao invés de Módulo de Classificação da Questão (*Question Classification Module*) ou Processamento da Questão (*Question Processing*), utiliza-se o termo Direcionamento da Resposta por este ser mais aderente a finalidade do módulo dentro do processamento global do ciclo de pergunta-resposta, isto é, o de interpretar a questão (e em alguns casos complementá-la) para direcionar o processo de coleta de informação e de definição da resposta.
- Coleta de Informação: este termo é compatível com o termo utilizado por Walke e Karale (2013), que na tradução literal é Módulo de Recuperação de Informação (*Information Retrieval Module*).
- Definição da Resposta: ao invés de Módulo de Extração da Informação (*Answer Extraction Module*) ou Extração da Resposta (*Answer Extraction*), utiliza-se o termo Definição da Resposta porque o verbo extrair representa “tirar, puxar (alguma coisa) para fora de algo que a contém ou do qual ela constitui uma parte”<sup>4</sup> enquanto que no processamento de perguntas mais complexas, como as que pedem comparações não realizadas até o momento ou que envolvem opiniões, ações como compor e formar são exigidas, sendo que, nestes casos, por exemplo, estas ações já não estariam representadas por Extração.

---

<sup>4</sup> <http://michaelis.uol.com.br/moderno/portugues/index.php>

A Figura 4 mostra os módulos arquiteturais sob a ótica dos componentes que os compõe. Ao contrário do que ocorre em relação aos módulos, não é obrigatória a adoção de 100% dos componentes. Assim como no caso dos módulos, também há um fluxo de processamento natural, indicado pelo vermelho escuro das setas. Porém, exceções a este fluxo podem ocorrer, sendo o fluxo alternativo marcado pelo vermelho claro das setas.

**Figura 4. Componentes arquiteturais**



**Fonte:** o autor

Ao trabalhar os componentes, é importante observar um conceito tradicional da área de comunicação. Este se chama ruído e gera interferências que prejudicam a comunicação de dados e informações. Perguntas com erros de pontuação ou de escrita, por exemplo, geram ruídos que afetam negativamente a eficácia do sistema. Fontes de dados com informações contraditórias ou que apresentam erros gramaticais frequentes também geram ruídos. O ruído é algo que emperra o funcionamento dos componentes de um QAS. Assim, ao olhar cada um destes componentes, é necessário avaliar o nível de ruído com o qual irão trabalhar. A

construção de estratégias que diminuam os ruídos pode ser necessária, como ocorre na implementação 5 exposta na Seção 2.2.

### **3.2.1. Direcionamento da Resposta**

O módulo Direcionamento da Resposta é dividido nos componentes: 1) Inserção da Pergunta, 2) Agregação de Elementos e Propriedades e 3) Qualificação das Variáveis Direcionadoras.

#### **3.2.1.1. Inserção da pergunta**

Este componente aciona elementos e tratamentos relacionados à interação homem-computador, utilizados tanto para inserção da pergunta quanto para obtenção de informações complementares junto ao usuário, caso estas últimas sejam necessárias. Exemplos de ações: carregamento de elementos visuais da interface, retorno de aviso de erro, ativação de ferramentas de acessibilidade, disponibilização de manuais de uso.

O componente Inserção da Pergunta não é tratado de forma específica pelas implementações descritas na Seção 2.2. Mas como qualquer sistema que deve interagir com um ser humano este é um componente de extrema importância.

#### **3.2.1.2. Agregação de elementos e propriedades**

Este componente agrupa elementos e propriedades respeitando o contexto no qual a pergunta está inserida. Exemplos de ações: vinculação de informações com base em perguntas realizadas anteriormente, complementação de informação por meio de base de conhecimento específico, cruzamento da pergunta com informações pessoais do usuário.

A Tabela 5 expõe como cada implementação apresentada na Seção 2.2 trabalha este componente. Observa-se que somente a implementação 2 acabou adotando mecanismos para o componente em questão. A causa disto é a baixa utilização do conceito de Contexto (Subseção 3.1.3) pelas implementações analisadas.

**Tabela 5. Implementações & Agregação de elementos e propriedades**

Sistemas	Agregação de elementos e propriedades
Implementação 1	N/D*
Implementação 2	Utiliza base de conhecimento específico do domínio com informações que permitem ao sistema entender como um humano interpretaria as informações apresentadas, inclusive permitindo predizer atributos não especificados, por exemplo, transformando conceitos qualitativos como “melhor” em conceitos mais palpáveis como preço, itens disponíveis no quarto etc.
Implementação 3	N/D*
Implementação 4	N/D*
Implementação 5	N/D*

\* N/D: Não há dados; artigo não cita desenvolvimento específico para o ponto analisado.

**Fonte:** o autor

### 3.2.1.3. Qualificação das variáveis direcionadoras

Este componente relaciona e classifica as variáveis que irão guiar a coleta de informação e a definição da resposta. Exemplos de ações: classificação da pergunta, definição e categorização dos termos de pesquisa, identificação de conjuntos de entidades, definição de termos chave.

A Tabela 6 apresenta as características do processamento de cada implementação que, analisadas de forma conjunta, geraram a definição e o nome do componente em questão.

**Tabela 6. Implementações & Qualificação das variáveis direcionadoras**

Sistema	Qualificação das variáveis direcionadoras
Implementação 1	Gera chave de três níveis: “tipo genérico: categorias: subcategoria”. São 7 tipos genéricos ligados aos pronomes interrogativos, 44 tipos de categoria (ex. local, data, nome próprio) e 110 subcategorias (ex. aeroporto, montanha, pessoa). Também analisa atributos como palavra-chave.
Implementação 2	Identifica os seguintes componentes da questão: Contexto, Expressão Comparativa ou Aferitiva, Entidade, Modo de Comparaçao e Restrição. O

	Contexto representa as demandas e necessidades do usuário, classificados em Itinerário, Acomodação, Transporte, Passeios, Tempo e Custo. A Expressão Comparativa ou Aferitiva ( <i>Comparative or Evaluative Expression</i> , CEE) identifica os termos de comparação ou aferição das Entidades. A Entidade representa os objetos a serem comparados. O Modo de Comparação representa os tipos possíveis de comparação. A Restrição representa critérios que guiam a comparação identificando restrições relacionadas às preferências do usuário. Este componente também envolve os Atributos Comparativos e Aferitivos ( <i>Comparative and Evaluative Features</i> , CEF), sendo estes de 17 tipos, como Local Para, Local De, Local Indispensável, Tempo de Idade, Tempo Limite, Detalhe do Viajante, Detalhe do Grupo, Orçamento, Adjetivo Modificador etc.
Implementação 3	Reconhece e tipifica os múltiplos fatos encontrados dentro da questão separando-as em subquestões, conforme regras e padrões léxico-sintáticos. Três padrões genéricos (Sub-Árvores Independentes, Unidades Combináveis e Segmento com Qualificadores) auxiliam nesta tarefa. As subquestões sofrem tratamentos heurísticos para evitar serem consideradas subquestões que não se aplicam ao caso. As subquestões têm sua relação definida como Paralela ou Aninhada, sendo que na dependência Paralela as subquestões são tratadas de forma independente e na dependência Aninhada as subquestões são processadas conforme ordem lógica determinada por qual subquestão "contém" a outra. Como a quebra em subquestões pode deixar algumas informações importantes de fora do escopo da subquestão, estas são reescritas de forma a garantir que informações contextuais chave sejam mantidas juntas às subquestões.
Implementação 4	Utiliza a biblioteca Stanford CoreNLP (SCNLP) para gerar uma árvore de dependência e POST com o objetivo de traduzir a pergunta em um modelo RDF de três elementos. A lista obtida é revisada com o objetivo de adaptar os termos à base ontológica da DBpedia. Também é definido o tipo de resposta esperada, utilizando a seguinte relação entre pronome interrogativo e tipo de resposta: Quem = Pessoa, Organização, Empresa; Aonde = Local; Quando = Data; Quanto = Número. Considera que no caso de questões que utilizam Qual, não é necessário haver a checagem do tipo de resposta devido à importância que o substantivo assume neste tipo de pergunta.
Implementação 5	Identifica o tipo de resposta esperada dentre 53 tipos, sendo que a definição ocorre por meio de um modelo de classificação <i>perceptron</i>

	multiclasses e da atribuição de uma série de características léxicas, sintáticas e semânticas. Estes tipos são reclassificados dentre sete Entidades reconhecíveis pelo NERC: Pessoa, Organização, Local, Medida, Número, Data e Hora. Também é identificado um conjunto de palavras-chave a ser utilizado nas <i>queries</i> de coleta de informação. Atrelado às palavras-chave, são designadas relevâncias heurísticas, sendo a pontuação fixa (ex. Adjetivos = 5, Entidades = 8).
--	---

\* N/D: Não há dados; artigo não cita desenvolvimento específico para o ponto analisado.

Fonte: o autor

### 3.2.2. Coleta de Informação

O módulo Coleta de Informação é dividido nos componentes: 1) Coordenação e adaptação das fontes de dados, 2) Pesquisa de dados e 3) Tratamento dos dados coletados.

#### 3.2.2.1. Coordenação e adaptação das fontes de dados

Este componente é composto por mecanismos que visam realizar o pré-processamento de informações com o objetivo de preparar e direcionar a coleta de informações. Exemplos de ações: definição das fontes de informação consultadas, indexação de termos chave, transformação de som em texto, inclusão de tags.

A Tabela 7 apresenta as características do processamento de cada implementação que, analisadas de forma conjunta, geraram a definição e o nome do componente em questão.

**Tabela 7. Implementações & Coordenação e adaptação das fontes de dados**

Sistema	Coordenação e adaptação das fontes de dados
Implementação 1	Armazena e indexa bases e trechos relevantes localmente.
Implementação 2	Mantém três tabelas locais: 1 – Localização Hierarquizada, contendo dados de cidade, estado, país, latitude, longitude, população, língua, tipo de turismo etc., 2 – Documentos Indexados, contendo dados de endereço de documentos, indexação direta e indexação indireta e 3 – Detalhes de Aeroportos e Ferrovias, contendo dados de cidade, aeroporto, código IANA, estação de trem, código de estação de trem. Estas são otimizadas

	para ler fontes de informação pré-definidas: WikiTravel, Lonely Planet, Kayak e Google Maps.
<b>Implementação 3</b>	N/D*
<b>Implementação 4</b>	Relaciona elementos da base ontológica da DBpedia dentro de uma estrutura RDF de três elementos.
<b>Implementação 5</b>	N/D*

\* N/D: Não há dados; artigo não cita desenvolvimento específico para o ponto analisado.

Fonte: o autor

### 3.2.2.2. Pesquisa de dados

Este componente escaneia as fontes de dados, respeitando o direcionamento dado pelo módulo Direcionamento da Resposta, para recuperar informações relevantes que alimentem a composição da resposta. Exemplos de ações: estruturação de queries de pesquisa, alimentação dos mecanismos de busca, recuperação dinâmica de informação.

A Tabela 8 apresenta as características do processamento de cada implementação que, analisadas de forma conjunta, geraram a definição e o nome do componente em questão.

**Tabela 8. Implementações & Pesquisa de dados**

Artigo	Pesquisa de dados
<b>Implementação 1</b>	Utiliza múltiplos agentes de busca como Indri IR Engine e Bing Search API. A coleta ocorre tanto em bases encontradas localmente (ex. AQUAINT Corpus, Wikipedia Corpus) quanto na Web.
<b>Implementação 2</b>	A informação do ponto turístico é utilizada para direcionar a pesquisa na tabela Documentos Indexados, sendo esta tabela utilizada para mapear os possíveis locais aonde a informação desejada pode ser encontrada. As informações das bases locais são cruzadas com as fontes WikiTravel, Lonely Planet, Kayak e Google Maps. A busca é otimizada pela utilização de HTML tags especiais, agrupamento de informações específicos ou RSS feed.
<b>Implementação 3</b>	São coletadas informações conforme modelo padrão estabelecido pelo sistema Watson, da IBM, que tem como característica ser de domínio

	aberto e ter inter-relação flexível com as fontes consultadas. O sistema Watson é abastecido por informações derivadas das subquestões obtidas no módulo de Definição da Resposta. Na dependência Paralela a coleta de informações por subquestão ocorre de forma paralela. Na dependência Aninhada primeiro são buscadas informações da subquestão "contida" e depois informações relacionadas à subquestão "contém", sendo que esta última já revisada com as informações da subquestão "contida". Também é coletado informações inserindo a pergunta integral, sem ser dividida em subquestões, no QAS Watson.
<b>Implementação 4</b>	São analisadas todas as possíveis combinações entre as informações existentes dentro da lista RDF de três elementos adaptada à ontologia da DBpedia para construir SPARQL <i>queries</i> utilizadas para ler a fonte DBpedia.
<b>Implementação 5</b>	Trabalha com dois tipos de modelos de coleta de informação, ambos utilizando o conjunto de palavras-chave identificadas no módulo Definição da Resposta. O primeiro processa o conjunto de palavras-chave estruturando <i>queries</i> dinâmicas que trabalham de forma a ajustar iterativamente o número de palavras-chave e a proximidade com que estas devem ser encontradas. A ferramenta de extração de informação utilizada é a Lucene IR (Apache). O segundo modelo é adaptado para transcrições e utiliza um mecanismo SDR ( <i>Spoken Document Retrieval</i> ). Trabalha com similaridade fonética para encontrar as palavras-chave, melhorando a interpretação dos trechos de documentos obtidos com a utilização de mecanismos ASR.

\* N/D: Não há dados; artigo não cita desenvolvimento específico para o ponto analisado.

**Fonte:** o autor

### 3.2.2.3. Tratamento dos dados coletados

Este componente agrupa classificações e propriedades às informações recuperadas, sendo estas utilizadas pelo módulo Definição da Resposta. Exemplos de ações: contagem do número de palavras chave, agregação de valores de frequência calculados sobre propriedades específicas, agregação de tags, identificação de entidades específicas.

A Tabela 9 apresenta as características do processamento de cada implementação que, analisadas de forma conjunta, geraram a definição e o nome do componente em questão.

**Tabela 9. Implementações & Tratamento dos dados coletados**

Sistema	Tratamento dos dados coletados
<b>Implementação 1</b>	Utiliza oito extratores de Entidades de forma integrada para identificar Entidades dentro dos trechos coletados.
<b>Implementação 2</b>	Trata as informações coletadas para identificar objetos de entidade e informações associadas a estes objetos. São identificados atributos conforme regras contidas na base de conhecimento específico do domínio. Os atributos das entidades são valorados, também seguindo padrões pré-estabelecidos pela base de conhecimento do domínio.
<b>Implementação 3</b>	Os atributos identificados nos trechos coletados são atrelados a índices de confiança, compondo valor que permite ranquear as informações obtidas.
<b>Implementação 4</b>	A cada query trabalhada é atrelada uma pontuação que se baseia na frequência obtida pelos predicados.
<b>Implementação 5</b>	São identificadas entidades dentro dos trechos coletados. O NERC utilizado atua em duas fases. A primeira é a Entity Recognition Model, que trabalha com a técnica BIO <i>tagging scheme</i> em que cada palavra de um trecho é classificada com B, I ou O. B ( <i>Beginning</i> ) ocorre quando é início de uma nova entidade. I ( <i>Inside</i> ) ocorre quando outra palavra está dentro da entidade. O ( <i>Outside</i> ) ocorre quando uma palavra não está atrelada a uma entidade. A segunda fase chama-se <i>Entity Classification Model</i> e trabalha as tags classificando-as dentro dos padrões pré-definidos de entidades (Pessoa, Organização, Local, Medida, Número, Data e Hora).

\* N/D: Não há dados; artigo não cita desenvolvimento específico para o ponto analisado.

**Fonte:** o autor

### 3.2.3. Definição da Resposta

O módulo Definição da Resposta é dividido nos componentes: 1) Constituição da resposta e 2) Exposição da resposta.

### 3.2.3.1. Constituição da resposta

Este componente processa o conjunto de informações gerado pelos módulos Direcionamento da Resposta e Coleta de Informação visando fazer a triagem, organização e tratamento dos trechos coletados originando a resposta para a pergunta realizada. Exemplos de ações: cálculo de pesos conforme correlação entre variáveis direcionadoras e características dos trechos coletados, correlacionamento entre o tipo de pergunta e entidades, análise de similaridade de resultados, ranqueamento conforme pontuações probabilísticas e heurísticas.

A Tabela 10 apresenta as características do processamento de cada implementação que, analisadas de forma conjunta, geraram a definição e o nome do componente em questão.

**Tabela 10. Implementações & Constituição da resposta**

Sistema	Constituição da resposta
Implementação 1	Utiliza gráfico sistêmico de objetos para armazenar características relevantes sobre entradas, modelos, saídas, nós de processamento e dependências relacionados ao processo de afunilamento para seleção e ranqueamento das potenciais respostas. Cruza as informações de entidades relacionadas aos trechos coletados com as classificações obtidas durante o módulo Direcionamento da Resposta.
Implementação 2	As pontuações dos atributos de cada objeto dos trechos coletados são somadas de forma ponderada levando-se em consideração os atributos identificados durante o módulo Direcionamento da Resposta.
Implementação 3	É trabalhada a recombinação dos trechos coletados, os sintetizando e ranqueando. É trabalhada de forma diferente as listas geradas pela coleta dos trechos quando consideradas as dependências Paralela e Aninhada. Na dependência Paralela as listas geradas pelas subquestões são unificadas, sendo a nova lista cruzada com a lista gerada pelo tratamento da pergunta não decomposta. Este cruzamento serve para rever os índices de confiança conforme peso de atributos heurísticos. No caso da dependência Aninhada, é cruzada a lista obtida por meio da coleta de informações por subquestão com a lista gerada pelo tratamento da pergunta não decomposta. A resposta final é classificada através da

	comparação entre os índices de confiança obtidos no tratamento das subquestões versus os índices de confiança obtidos no tratamento da pergunta não decomposta.
<b>Implementação 4</b>	São checados o alinhamento entre as características dos dados coletados e o tipo de resposta esperada. Os candidatos alinhados são ranqueados conforme a pontuação de frequência.
<b>Implementação 5</b>	São cruzadas as entidades encontradas no modulo Coleta de Informação com as entidades identificadas no módulo de Definição da Resposta. Os trechos aderentes são ranqueados por meio de uma junção de medidas, sendo que os dois principais modelos de pontuação tem base heurística, um medindo a distância e densidade das palavras-chave e outro tratando a relação de similaridade sintática entre trechos e pergunta.

\* N/D: Não há dados; artigo não cita desenvolvimento específico para o ponto analisado.

**Fonte:** o autor

### 3.2.3.2. Exposição da resposta

Este componente contém regras relacionadas aos elementos de interação homem-computador voltadas à exposição da resposta. Exemplos de ações: indicação de resposta não obtida, definição do número de respostas dispostas na tela, transformação de texto em som.

A Tabela 11 apresenta as características do processamento de cada implementação que, analisadas de forma conjunta, geraram a definição e o nome do componente em questão. As descrições evidenciam uma falta de atenção por parte dos autores das implementações em relação a este componente. Porém, este não deixa de ser menos importante no momento em que está atrelado a mecanismos que favorecem a interação homem-máquina.

**Tabela 11. Implementações & Exposição da resposta**

Sistema	Exposição da resposta
<b>Implementação 1</b>	Mostra a resposta posicionada em primeiro lugar no ranqueamento.
<b>Implementação 2</b>	Mostra as cinco respostas que apresentam maior pontuação dentro do ranqueamento.
<b>Implementação 3</b>	Mostra a resposta posicionada em primeiro lugar no ranqueamento.

<b>Implementação 4</b>	Mostra a resposta posicionada em primeiro lugar no ranqueamento.
<b>Implementação 5</b>	Mostra a resposta posicionada em primeiro lugar no ranqueamento.

\* N/D: Não há dados; artigo não cita desenvolvimento específico para o ponto analisado.

Fonte: o autor

### 3.3. Aprendizado

O objetivo desta seção é a de identificar se técnicas de aprendizado de máquina (*machine learning*) são utilizadas pelas implementações expostas na Seção 2.2. Este se deve à tendência cada vez mais presente de trabalhar com QAS que utilizam este tipo de mecanismo.

Aprendizado de máquina é um dos inúmeros mecanismos que podem ser adotados dentro dos componentes dos módulos da arquitetura de um QAS. Este mecanismo é destacado frente aos tantos outros existentes porque um QAS apresenta cenários de utilização e de composição de respostas tão diversos que podem ser considerados ilimitados. Ao utilizar mecanismos de aprendizado de máquina, espera-se obter um ganho constante na capacidade de geração de respostas válidas dentro dos diversos cenários conforme a ferramenta seja utilizada sem a necessidade de intervenções manuais.

Na Tabela 12, para cada implementação apresentada durante a Seção 2.2, é identificado se foram utilizados mecanismos de aprendizado e, caso a resposta seja afirmativa, é dada uma breve descrição do foco destes.

**Tabela 12. Implementações & Aprendizado**

Sistema	Apresenta Mecanismos de Aprendizado?	Descrição
<b>Implementação 1</b>	Sim	Utiliza o gráfico sistêmico de objetos para calibrar pesos de características dos objetos.
<b>Implementação 2</b>	Não	-
<b>Implementação 3</b>	Sim	Trabalha sobre os atributos utilizados para gerar o índice de confiança dos trechos coletados, avaliando a pontuação final obtida pelos trechos

		frente ao resultado final esperado, o que permite recalibrar os pesos dos atributos na pontuação caso necessário.
Implementação 4	Não	-
Implementação 5	Sim	Utilizado para ajustar o processo de reconhecimento de entidades (NERC) e no modelo de ranqueamento tomando como base o ajuste combinatório das avaliações heurísticas realizadas.

Fonte: o autor

### 3.4. Considerações do Capítulo

A construção de um QAS envolve estruturas básicas bem definidas em seu nível mais alto de abstração. Qualquer sistema que pretenda ser um QAS tem de ser capaz de interpretar uma pergunta, coletar informações e definir a melhor resposta. Sem alguma destas ações é impossível atender ao objetivo de extrair conteúdo conciso e relevante de uma ou mais fontes de informação em resposta a uma questão realizada.

Neste capítulo foram introduzidos conceitos fundamentais relacionados a qualquer QAS, que foram agrupados dentro de Identidade e Arquitetura. A relação entre estes dois conceitos é apresentada pela Figura 5.

Figura 5. Identidade & Arquitetura



Fonte: o autor

Os módulos da arquitetura, representados pelos retângulos, contêm os componentes que tornam vivo um sistema de QAS. Há uma grande gama de mecanismos que pode ser utilizada em cada um dos componentes de cada um dos módulos arquiteturais e a correlação entre eles pode ser estabelecida de diversas formas. Assim, é aberto um leque enorme de opções para construir um QAS. Em um cenário como este, é recomendável ter uma visão clara das necessidades que devem ser atendidas pelo sistema, auxiliando na seleção dos mecanismos a serem empregados. E para compor a visão das necessidades toma-se como direcionador as características definidas em Identidade, representada pelos quadrados encontrados no centro da Figura 5.

As características da Identidade devem ser olhadas em conjunto, isto é, quando for definir ou modificar uma característica, todas as outras também devem ser analisadas. Muitas vezes, um efeito desejado somente é obtido quando há um alinhamento específico entre as características. Por exemplo, Bhaskar, Pal e

Bandyopadhyay (2012) limitam o Escopo e criam uma rigidez no Acesso de seu sistema abrindo espaço para a implantação de mecanismos avançados de Contexto e também possibilitando o tratamento de Perguntas mais complexas, se comparado aos outros sistemas descritos durante o Capítulo 2.

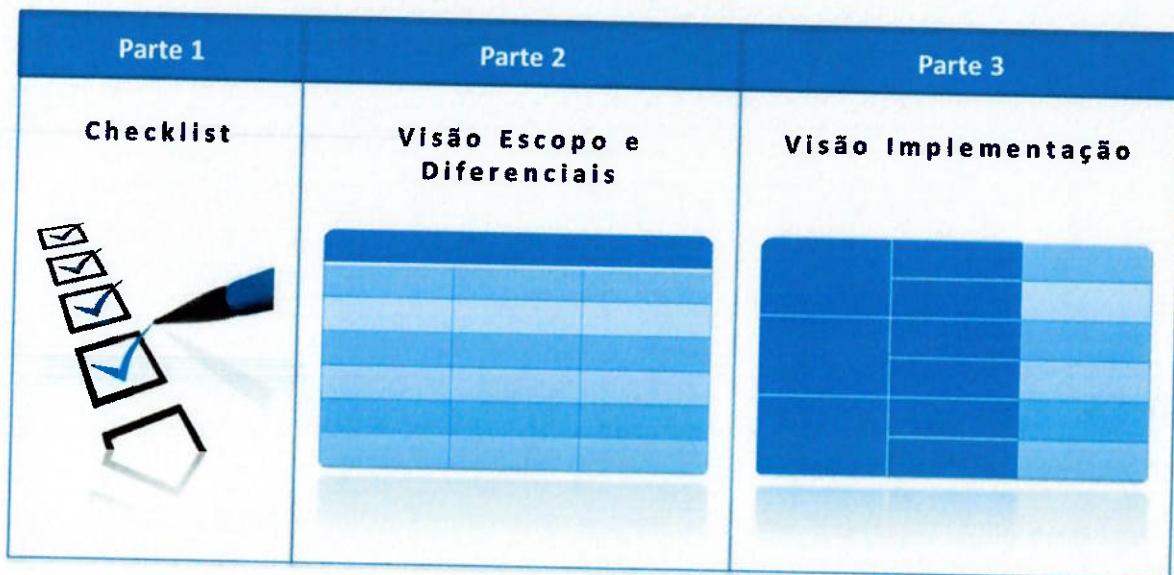
Neste capítulo também foi tecido um breve comentário sobre a técnica de aprendizado de máquina. Esta técnica chama cada vez mais a atenção por permitir diminuir a codificação manual, o que agiliza a adaptação de um sistema a novas situações. Como pode ser considerada infinita a quantidade de perguntas capazes de serem formuladas e ainda por cima é necessário cruzá-las ao imenso mundo de informações disponíveis, uma técnica como esta, que auxilia na adaptação não manual de um determinado sistema, é muito bem-vinda. Porém, ressalte-se, esta é mais uma das diversas técnicas que podem ou não ser utilizadas na construção de QAS, assim como NERC, POST, avaliações estatísticas de frequência etc.

Utilizando como alicerce as informações apresentadas neste capítulo, segue-se para o Capítulo 4, no qual é estruturado um guia para seleção de sistemas do tipo pergunta-proposta, objetivo principal do atual estudo.

#### 4. GUIA DE SELEÇÃO

Este capítulo apresenta o guia de seleção e simulações de utilização. O objetivo do guia é o de direcionar a avaliação de QAS e o de permitir a escolha de um que mais se alinhe às necessidades particulares de um avaliador. O guia é dividido em três partes, como mostrado na Figura 6.

**Figura 6. Guia de seleção**



Fonte: o autor

O *Checklist* apresenta um conciso conjunto de necessidades filtrando de forma rápida sistemas com capacidade de atender ao avaliador. A Visão Escopo e Diferenciais apresenta a abrangência e particularidades dos sistemas elegíveis conforme instanciação do *Checklist*. A Visão Implementação é instanciada caso o usuário tenha interesse em entender melhor a rotina de processamento dos sistemas que lhe interessem. Cada uma das partes apresentadas é detalhada nas próximas seções.

##### 4.1. Checklist

O *Checklist* é estruturado sobre as bases conceituais apresentadas no Capítulo 3. Leva em consideração um conjunto de características que dão direção para o arranjo de elementos básicos de processamento que compõem um QAS particular. A Figura 7 apresenta o *Checklist*.

Figura 7. Checklist

## Checklist

---

**1. Quais os tipos de pergunta que o sistema deve ser capaz de responder?**

- Fato
- Fato Composto
- Comparação
- Aferição
- Nenhuma das alternativas

**2. O sistema deve ser capaz de verificar se há informações subjetivas e/ou incompletas na pergunta, apresentando mecanismos que permitam dar objetividade e/ou complementar as informações fornecidas?**

- Sim, obrigatoriamente
- Não é necessário

**3. O sistema deve ser capaz de trabalhar com um encadeamento de perguntas de forma que perguntas anteriores forneçam informações imprescindíveis para a interpretação da próxima pergunta ou para a composição de sua resposta?**

- Sim, obrigatoriamente
- Não é necessário

**4. O sistema deve estar preparado para interpretar questões ou coletar informações estruturadas em formato pouco aderente às regras gramaticais e normas cultas que regem a composição de textos?**

- Sim, obrigatoriamente
- Não é necessário

**5. O sistema deve ser flexível para encontrar informações em fontes não pré-definidas e diversificadas ou em fontes nas quais os dados não seguem um padrão de organização e classificação?**

- Sim, obrigatoriamente
- Não é necessário

---

Fonte: o autor

Abaixo são detalhados os itens do Checklist.

***ITEM 1: “Quais os tipos de pergunta que o sistema deve ser capaz de responder?”***

Este item está relacionado com a característica Pergunta. Como visto na Subseção 3.1.1, a Pergunta exerce alto impacto na definição dos meios lógicos de interpretação e de processamento em um QAS. O conjunto de tipos de Pergunta utilizado para compor as alternativas deste item pode sofrer acréscimo caso novos sistemas tragam tipos de Pergunta diferentes aos dos utilizados atualmente pelo Checklist. Considerações sobre o mapeamento dos tipos de Pergunta são feitas na Subseção 3.1.1.

Uma ou mais alternativas podem ser selecionadas dentro deste item. A marcação da alternativa “Nenhuma das alternativas” indica que nenhum dos sistemas avaliados até o momento atende às necessidades do usuário.

***ITEM 2: “O sistema deve ser capaz de verificar se há informações subjetivas e/ou incompletas na pergunta, apresentando mecanismos que permitam dar objetividade e/ou complementar as informações fornecidas?”***

Este item está relacionado com as características Contexto (ver Subseção 3.1.3) e Escopo (ver Subseção 3.1.2). Aqui é verificado se o sistema é capaz de resolver subjetividades ou indefinições trazidas dentro da questão. Um exemplo é a pergunta: Nasci em 01/01/2000, qual é a minha idade? O sistema, neste caso, precisa estar preparado para calcular a idade e interpretar que a data final de referência é “hoje”. A data final de referência depende do dia em que a pergunta foi realizada (Contexto) e o sistema deve estar preparado para abordar assuntos relacionados à “Data” (Escopo), incluindo métodos de cálculo.

Somente uma alternativa pode ser selecionada para este item. A alternativa “Sim, obrigatoriamente” indica que a existência da funcionalidade é fundamental para o usuário, desclassificando sistemas que não a contenham. A alternativa “Não é necessário” serve como contraponto à alternativa “Sim, obrigatoriamente”, sendo que esta não desclassifica sistemas ao indicar que a existência da funcionalidade avaliada não é obrigatória.

**ITEM 3: “O sistema deve ser capaz de trabalhar com um encadeamento de perguntas de forma que perguntas anteriores forneçam informações imprescindíveis para a interpretação da próxima pergunta ou para a composição de sua resposta?”**

Este item está relacionado com a característica Contexto (ver Subseção 3.1.3). Aqui é verificado se o sistema é capaz de relacionar um conjunto particular e temporário de perguntas e respostas. Por exemplo, a pergunta “Ele está perto?” pode ser derivada da pergunta “João está vindo?”. Neste caso, a palavra “Ele” representa “João”. Somente para efeito de comparação, caso não haja pergunta anterior a “João está vindo?” seria necessário predizer que “João” é o marido da usuária, mecanismo que estaria atrelado ao item 2 do Checklist.

Somente uma alternativa pode ser selecionada para este item. A alternativa “Sim, obrigatoriamente” indica que a existência da funcionalidade é fundamental para o usuário, desclassificando sistemas que não a contenham. A alternativa “Não é necessário” serve como contraponto à alternativa “Sim, obrigatoriamente”, sendo que esta não desclassifica sistemas ao indicar que a existência da funcionalidade avaliada não é obrigatória.

**ITEM 4: “O sistema deve estar preparado para interpretar questões ou coletar informações estruturadas em formato pouco aderente às regras gramaticais e normas cultas que regem a composição de textos?”**

Este item está relacionado com o conceito de ruído (ver Seção 3.2). A aderência de perguntas e fontes de dados à norma gramatical culta diminui a presença de ruídos, favorecendo o funcionamento dos componentes do QAS. Caso seja previsto que o QAS trabalhará com perguntas e fontes de dados que apresentem ruídos frequentes, é recomendável que o sistema contenha mecanismos específicos para tratamento destes.

Somente uma alternativa pode ser selecionada para este item. A alternativa “Sim, obrigatoriamente” indica que a existência da funcionalidade é fundamental para o usuário, desclassificando sistemas que não a contenham. A alternativa “Não é

“necessário” serve como contraponto à alternativa “Sim, obrigatoriamente”, sendo que esta não desclassifica sistemas ao indicar que a existência da funcionalidade avaliada não é obrigatória.

**ITEM 5: “O sistema deve ser flexível para encontrar informações em fontes não pré-definidas e diversificadas ou em fontes nas quais os dados não seguem um padrão de organização e classificação?”**

Este item está relacionado com a característica Acesso (ver Subseção 3.1.4). A descrição do item remete ao Acesso Flexível. Pretende avaliar se o usuário necessita obter informações de uma grande gama de fontes ou de fontes não estruturadas.

Somente uma alternativa pode ser selecionada para este item. A alternativa “Sim, obrigatoriamente” indica que o sistema deve apresentar o Acesso Flexível. A alternativa “Não é necessário”, contraponto da alternativa “Sim, obrigatoriamente”, indica que o Acesso pode ser tanto Rígido quanto Flexível.

A Tabela 13 apresenta resumidamente o objetivo e a forma de seleção dos itens do Checklist.

**Tabela 13. Resumo Checklist**

Checklist	Objetivo	Seleção de Alternativas
Item 1	Mapeia o tipo de Pergunta.	Uma ou mais.
Item 2	Verifica se o sistema é capaz de resolver subjetividades ou indefinições trazidas dentro da questão.	Apenas uma.
Item 3	Verifica se o sistema contém mecanismos para correlacionar um encadeamento de informações derivadas de perguntas sequenciais.	Apenas uma.
Item 4	Verifica se o sistema apresenta mecanismos de tratamento de ruídos.	Apenas uma.
Item 5	Verifica se o Acesso é do tipo Flexível.	Apenas uma.

**Fonte:** o autor

Em um cenário no qual o usuário marque os itens 2 a 5 do Checklist com a resposta “Sim, obrigatoriamente”, todas as funcionalidades avaliadas se cruzam com o tipo de Pergunta demarcado no item 1, reduzindo o funil de seleção dos sistemas. No cenário oposto, em que o usuário marque os itens 2 a 5 com a resposta “Não é necessário”, o usuário resume sua seleção pelo tipo de Pergunta demarcado no item 1 do Checklist, afrouxando o funil de seleção.

Passados todos os itens do *Checklist*, os sistemas que atendem as necessidades apontadas pelo avaliador são listados dentro da estrutura da Visão Escopo e Diferenciais. Caso o conjunto de alternativas selecionadas dentro do *Checklist* elimine todos os sistemas disponíveis, não é instanciada a Visão Escopo e Diferenciais.

#### 4.2. Visão Escopo e Diferenciais

A Visão Escopo e Diferenciais relaciona os sistemas que estão em conformidade com as alternativas selecionadas dentro do *Checklist*. A tabela traz junto a esses sistemas informações sobre o Escopo (Subseção 3.1.2) e uma breve descrição relacionando os diferenciais de cada um. A Tabela 14 exibe os campos da tabela Visão Escopo e Diferenciais.

**Tabela 14. Visão Escopo e Diferenciais**

Sistema	Escopo	Diferenciais
<i>Nome do sistema.</i>	<i>Restrição (Assunto,...,Assunto).</i>	<i>Descrição dos diferenciais.</i>

**Fonte:** o autor

A escolha dos campos que compõe esta visão leva em consideração a premissa de objetividade de comparação. Este modelo permite realizar uma rápida análise comparativa entre os sistemas que acabam por apresentar características alinhadas às necessidades apontadas dentro do *Checklist*. O campo Escopo é eliminatório, já que direciona o que é passível de ser respondido pelo QAS analisado. Caso o avaliador considere que os termos inseridos neste campo para um sistema específico não se alinham as suas necessidades, o sistema deve ser descartado. O

campo Diferenciais resume características que permitem particularizar aqueles QAS que não foram desclassificados nem pelo *Checklist*, nem pelo campo Escopo.

A partir da Visão Escopo e Diferenciais podem ser acessadas informações específicas sobre a rotina de processamento de cada sistema, caso ainda restem dúvidas sobre qual é o melhor. Estas informações são estruturadas sob a ótica da Visão Implementação.

#### 4.3. Visão Implementação

A Visão Implementação traz informações sobre um QAS organizadas em blocos de processamento de propósito específico. A construção desta visão toma como base a perspectiva arquitetural apresentada na Seção 3.2. Somada a esta perspectiva, é inserido um último campo para identificar se há mecanismos relacionados a técnicas de aprendizado de máquina e, se houver, descrevê-los brevemente (ver Seção 3.3). A Tabela 15 exibe a estrutura da Visão Implementação.

**Tabela 15. Visão Implementação**

Sistema	Nome do sistema.	
Direcionamento da Resposta	Inserção da pergunta	Descrição do processamento.
	Agregação de elementos e propriedades	Descrição do processamento.
	Qualificação das variáveis direcionadoras	Descrição do processamento.
Coleta de Informação	Coordenação e adaptação das fontes de dados	Descrição do processamento.
	Pesquisa de dados	Descrição do processamento.
	Tratamento dos dados coletados	Descrição do processamento.
Definição da Resposta	Constituição da resposta	Descrição do processamento.
	Exposição da resposta	Descrição do processamento.
Aprendizado		Descrição do processamento.

**Fonte:** o autor

Esta tabela pode relacionar um ou mais sistemas, conforme desejado pelo avaliador. Caso sejam selecionados dois sistemas, por exemplo, a Visão Implementação aparece no formato apresentado pela Tabela 16.

**Tabela 16. Visão Implementação – Seleção de 2 sistemas**

Sistema		Sistema 1	Sistema 2
Direcionamento da Resposta	Inserção da pergunta	Descrição.	Descrição.
	Agregação de elementos e propriedades	Descrição.	Descrição.
	Qualificação das variáveis direcionadoras	Descrição.	Descrição.
Coleta de Informação	Coordenação e adaptação das fontes de dados	Descrição.	Descrição.
	Pesquisa de dados	Descrição.	Descrição.
	Tratamento dos dados coletados	Descrição.	Descrição.
Definição da Resposta	Constituição da resposta	Descrição.	Descrição.
	Exposição da resposta	Descrição.	Descrição.
Aprendizado		Descrição.	Descrição.

**Fonte:** o autor

Assim como no caso do aprendizado de máquina, outras técnicas podem ser adicionadas ao final da tabela. Comas, Turmo e Márquez (2012) moldam uma tabela comparativa de sistemas na qual incluem um campo descritivo exclusivo para a técnica NERC, por exemplo. Porém, evitou-se expandir excessivamente a Visão Implementação, sendo escolhido o aprendizado de máquina devido à atual importância dada a este tipo de técnica.

#### 4.4. Aplicação

Esta seção cria um cenário de aplicação para o guia de seleção. Com isto é possível observar a dinâmica existente entre as diversas etapas do guia. As respostas geradas tomam como base informações obtidas pela análise das implementações

apresentadas na Seção 2.2. A Figura 8 mostra a primeira etapa, que é a seleção das alternativas do *Checklist*.

**Figura 8. Aplicação – Checklist**

## Checklist

---

**1. Quais os tipos de pergunta que o sistema deve ser capaz de responder?**

- Fato
- Fato Composto
- Comparação
- Aferição
- Nenhuma das alternativas

**2. O sistema deve ser capaz de verificar se há informações subjetivas e/ou incompletas na pergunta, apresentando mecanismos que permitam dar objetividade e/ou complementar as informações fornecidas?**

- Sim, obrigatoriamente
- Não é necessário

**3. O sistema deve ser capaz de trabalhar com um encadeamento de perguntas de forma que perguntas anteriores forneçam informações imprescindíveis para a interpretação da próxima pergunta ou para a composição de sua resposta?**

- Sim, obrigatoriamente
- Não é necessário

**4. O sistema deve estar preparado para interpretar questões ou coletar informações estruturadas em formato pouco aderente às regras gramaticais e normas cultas que regem a composição de textos?**

- Sim, obrigatoriamente
- Não é necessário

**5. O sistema deve ser flexível para encontrar informações em fontes não pré-definidas e diversificadas ou em fontes nas quais os dados não seguem um padrão de organização e classificação?**

- Sim, obrigatoriamente
- Não é necessário

A Tabela 17 mostra a segunda etapa, que é a instanciação da Visão Escopo e Diferenciais, mostrando os sistemas que se alinham ao conjunto de alternativas assinaladas dentro do *Checklist*.

**Tabela 17. Aplicação – Visão Escopo e Diferenciais**

Sistema	Escopo	Diferenciais
<b>Implementação 1</b>	(Data, Local, Pessoa, Organização)	Estímulo à propagação dos resultados obtidos em diversas etapas do QAS, impactando positivamente na performance do modelo de ranqueamento e, consequentemente, na seleção da resposta.
<b>Implementação 3</b>	(Data, Hora, Local, Endereço, Pessoa, Contatos Pessoais, Endereços Web, Organização, Número, Animal, Doença, Produto, Transporte, Clima, Acontecimento)	Trabalha integrado ao QAS Watson, da IBM, respondendo a perguntas que apresentam fatos múltiplos que devem ser correlacionados entre si.
<b>Implementação 5</b>	(Data, Hora, Local, Organização, Pessoa, Número, Medida)	Otimizado para ler fontes de origem oral, utiliza técnicas para dirimir pontos que possam prejudicar a interpretação das fontes consultada e, além disso, emprega o conceito <i>Dynamic Query Relaxation Procedure</i> que ajusta o modelo de query iterativamente até a qualidade da coleta de informação se tornar satisfatória.

**Fonte:** o autor

A Tabela 18 mostra a terceira etapa do guia. Tem como premissa que o avaliador quer saber mais informações sobre a Implementação 5, selecionando este sistema dentro da Visão Escopo e Diferenciais.

**Tabela 18. Aplicação – Visão Implementação**

Sistema		Implementação 5
Direcionamento da Resposta	Inserção da pergunta	--
	Agregação de elementos e propriedades	--
	Qualificação das variáveis direcionadoras	Identifica o tipo de resposta esperada dentre 53 tipos, sendo que a definição ocorre por meio de um modelo de classificação <i>perceptron</i> multiclasse e da atribuição de uma série de características léxicas, sintáticas e semânticas. Estes tipos são reclassificados dentre sete Entidades reconhecíveis pelo NERC: Pessoa, Organização, Local, Medida, Número, Data e Hora. Também é identificado um conjunto de palavras-chave a ser utilizado em <i>queries</i> . Atrelado às palavras-chave, são designadas relevâncias heurísticas, sendo a pontuação fixa (ex. Adjetivos = 5, Entidades = 8).
Coleta de Informação	Coordenação e adaptação das fontes de dados	--
	Pesquisa de dados	Trabalha com dois modelos, ambos utilizando o conjunto de palavras-chave identificadas no módulo Direcionamento da Resposta. O primeiro modelo processa o conjunto de palavras-chave estruturando <i>queries</i> dinâmicas que trabalham de forma a ajustar iterativamente o número de palavras-chave e a proximidade com que estas devem ser encontradas. Este modelo é apoiado pela ferramenta Lucene IR (Apache). O segundo modelo é adaptado para transcrições e utiliza um mecanismo SDR ( <i>Spoken Document Retrieval</i> ). Trabalha com similaridade fonética para encontrar as palavras-chave, melhorando a interpretação de documentos obtidos com a utilização de mecanismos ASR.
	Tratamento dos dados coletados	Os trechos coletados são analisados para identificação de entidades. O NERC utilizado atua em duas fases. A primeira chama-se <i>Entity Recognition Model</i> , que trabalha com a

		técnica BIO <i>tagging scheme</i> em que cada palavra de um trecho é classificada com B, I ou O. B ( <i>Beginning</i> ) ocorre quando é início de uma nova entidade. I ( <i>Inside</i> ) ocorre quando outra palavra está dentro da entidade. O ( <i>Outside</i> ) ocorre quando uma palavra não está atrelada a uma entidade. A segunda fase chama-se <i>Entity Classification Model</i> e trabalha as tags classificando-as dentro de padrões pré-definidos de entidades (Pessoa, Organização, Local, Medida, Número, Data e Hora).
<b>Definição da Resposta</b>	<b>Constituição da resposta</b>	As entidades relacionadas aos trechos tratados dentro do módulo Coleta de Informação são cruzadas com as entidades encontradas no módulo Direcionamento da Resposta. Os trechos que apresentam entidades aderentes entre os módulos são ranqueados por meio de uma junção de medidas, sendo que os dois principais modelos de pontuação tem base heurística, um medindo a distância e densidade das palavras-chave e outro tratando a relação de similaridade sintática entre trechos e pergunta.
	<b>Exposição da resposta</b>	Mostra a resposta posicionada em primeiro lugar no ranqueamento.
<b>Aprendizado</b>		Utilizado para ajustar o processo de reconhecimento de entidades (NERC) e no modelo de ranqueamento tomando como base o ajuste combinatório das avaliações heurísticas realizadas.

Fonte: o autor

Com a instanciação da Visão Implementação, o guia de seleção proposto se encerra. Porém, ainda é apresentada a Tabela 19 com o objetivo de mostrar a dinâmica de seleção de sistemas conforme são alteradas as alternativas do *Checklist*. São apresentados 16 cenários, sendo que o cenário 1 é o utilizado na Figura 8. Um conjunto específico de alternativas, independentemente se todos os itens foram checados, pode eliminar todos os sistemas disponíveis, situação indicada com “--”. A seleção dos cenários respeita as seguintes regras: 1) somente alternativas de um item podem ser modificadas entre cenários subsequentes e 2) os cenários não podem se repetir. O Apêndice C permite obter os resultados para todas as possíveis combinações do *Checklist*.

**Tabela 19. Cenários de resposta – Checklist**

Cenário	Item 1	Item 2	Item 3	Item 4	Item 5	Sistemas Selecionados
1	Fato	Não é necessário	Não é necessário	Não é necessário	Sim, obrigatoriamente	Implementações 1, 3 e 5
2	Fato	Não é necessário	Não é necessário	Sim, obrigatoriamente	Sim,	Implementação 5
3	Fato; Fato Composto	Não é necessário	Não é necessário	Sim, obrigatoriamente	--	--
4	Fato; Fato Composto	Não é necessário	Não é necessário	Sim, obrigatoriamente	--	--
5	Fato; Fato Composto	Sim, obrigatoriamente	--	--	Sim, obrigatoriamente	Implementação 3
6	Comparação; Aferição	Sim, obrigatoriamente	Não é necessário	Não é necessário	Não é necessário	Implementação 2
7	Comparação; Aferição	Sim, obrigatoriamente	Sim, obrigatoriamente	--	--	--
8	Fato	Sim, obrigatoriamente	--	--	--	--
9	Fato	Não é necessário	Sim, obrigatoriamente	--	--	--
10	Fato	Não é necessário	Não é necessário	Não é necessário	Não é necessário	Implementações 1, 3, 4 e 5
11	Fato	Não é necessário	Não é necessário	Sim, obrigatoriamente	Não é necessário	Implementação 5
12	Fato Composto	Não é necessário	Não é necessário	Sim, obrigatoriamente	--	--
13	Fato Composto	Não é necessário	Não é necessário	Não é necessário	Não é necessário	Implementação 3
14	Fato Composto	Não é necessário	Não é necessário	Não é necessário	Sim, obrigatoriamente	Implementação 3
15	Nenhuma das alternativas	--	--	--	--	--
16	Fato; Fato Composto; Comparação; Aferição	--	--	--	--	--

Fonte: o autor.

#### 4.5. Considerações do Capítulo

Por meio do guia de seleção proposto é possível agilizar a avaliação de um conjunto de QAS visando buscar o sistema que melhor se encaixe nas necessidades particulares de um determinado usuário. Isto é possível devido à seleção criteriosa dos direcionadores e do formato, que foram ganhando corpo ao se entender as características essenciais dos QAS. Durante o desenvolvimento do modelo, não foi esquecido de que esta área está em plena evolução e que, por isto, a introdução de novos sistemas na base de sistemas avaliados deve ser facilitada. O conjunto de conceitos, direcionadores e formatos contidos no atual estudo favorecem esta atualização.

Apesar do número reduzido de itens incluídos dentro do *Checklist*, a Tabela 19 mostra que existe dinamismo na seleção dos sistemas conforme são alteradas as alternativas. E na mesma tabela também há outro fato interessante: a presença constante de cenários em que o conjunto de alternativas elimina todos os sistemas. Esta eliminação ocorre oito vezes dentro dos dezesseis cenários construídos. Este fato pode ser atribuído a dois fatores: 1) número reduzido de sistemas avaliados e 2) os sistemas são limitados para atender ao conjunto de necessidades mapeadas. A quantidade, a complexidade e a especificidade dos mecanismos envolvidos na estruturação de um QAS acabam por forçá-los a atender a objetivos bem delineados. Assim, formar uma grande base de sistemas avaliados é necessário para diminuir os casos em que um conjunto de alternativas eliminate todos os sistemas. Este ponto reforça a importância de se ter um guia de seleção que favoreça a evolução da base de sistemas avaliados.

## 5. CONSIDERAÇÕES FINAIS

Qualquer QAS apresenta o mesmo objetivo: extrair conteúdo conciso e relevante de fontes diversas em resposta a questões feitas em linguagem natural. Eles atraem a atenção porque empregam a ideia de sumarização linguística, capacitando máquinas a gerar respostas concisas e relevantes respeitando valores, regras e relacionamentos linguísticos tipicamente humanos. Com isto, ajudam a resolver o problema da crescente discrepância entre o rápido crescimento da quantidade de informação disponível e a praticamente estável capacidade cognitiva e de processamento de informação dos homens.

Foi identificada a existência de modelos completos de QAS em diversas fontes acadêmicas. E a estas se somam modelos comerciais como o Siri, da Apple, o Cortana, do Windows, e o Google Now, do Google, que apresentam soluções de QAS integradas. O QAS Watson, da IBM, ganhou destaque ao ganhar o Jeopardy, um quiz show. Porém, não há indícios de um QAS consolidado que seja amplamente utilizado no dia-a-dia, o que nos leva a perguntar: O que restringe a disseminação e utilização de uma solução QAS?

Entender a essência dos modelos atuais de QAS ajuda a responder a esta pergunta. Para solucionar cada um dos desafios desta área, foram desenvolvidos algoritmos complexos e especializados que podem ser integrados de formas diversas. Somado a isso, para simplificar a estruturação de QAS, há uma tendência de se limitar o seu escopo. Assim geram-se diversos modelos específicos de QAS e estes são difíceis de serem compreendidos. O presente estudo ajuda a resolver esta dificuldade.

Ainda há muitos desafios relacionados aos sistemas do tipo pergunta-resposta. Como adaptar um QAS ao contexto do usuário? Como é a forma mais eficaz de troca de informações entre QAS e usuário, melhorando a efetividade das respostas geradas? Como os modelos mentais de pensamento de pessoas diferentes afetam a composição de respostas e como estes podem ser aproveitados em sistemas de QAS? Um dia um único QAS poderá gerar respostas para 100% das perguntas? E este mesmo QAS poderá alcançar uma taxa de 100% de respostas corretas? Verifica-se por meio destas perguntas o quanto de espaço há para a realização de

interessantes estudos na área. E estes estudos podem se aproveitar do modelo de avaliação aqui construído, agilizando a compreensão de QAS existentes, o que ajuda na compreensão de suas capacidades e limites.

### 5.1. Contribuições do Trabalho

Criou-se um guia de seleção prático e dinâmico para a compreensão das variáveis essenciais de qualquer sistema do tipo pergunta-resposta auxiliando e agilizando a seleção de um QAS específico que atenda as necessidades apontadas por um determinado usuário.

Para apoiar a construção do guia foram construídos conceitos, definições e modelos dentro do Capítulo 3. Estes também são contribuições valiosas do trabalho. Como exemplo, pode ser citada a Figura 2, a Figura 3, a Figura 4 e a Figura 5 e os conceitos que as cercam. O Capítulo 3, inclusive, pode ser utilizado em conjunto com o Capítulo 2, o Apêndice A e o Apêndice B para direcionar a estruturação de novos QAS.

O Capítulo 2 traz implementações concretas, o Capítulo 3 traz a essência dos QAS e o Capítulo 4 mostra como avaliar implementações. Este conjunto é uma boa introdução para qualquer pessoa que queira se aventurar no mundo dos QAS.

### 5.2. Trabalhos Futuros

Os posicionamentos e questionamentos do presente trabalho abrem espaço para uma série de estudos futuros. Exemplificando alguns deles:

- Avaliar novos sistemas de QAS a partir da estrutura sugerida pelo presente trabalho, visando evoluir no mapeamento dos modelos e dos mecanismos existentes, o que permite avançar na compreensão das potencialidades e necessidades da área.
- Transformar os componentes descritos na Figura 4 em uma visão Orientada a Objeto ou Orientada a Serviço, permitindo avançar na identificação de melhorias e a formatar um framework genérico para soluções de QAS.

- Aprofundar no entendimento de como um ser humano trabalha o ciclo de pergunta e resposta e cruzar este entendimento com o que existe hoje na área de QAS, mapeando as vantagens e desvantagens tanto em relação ao ser humano quanto em relação à máquina, resultando em maior clareza sobre as potencialidades e os limites da área de QAS.
- Construir e examinar cenários de interação ideais entre homem-máquina voltados a QAS e cruzar estes cenários com a interação oferecida pelos modelos existentes para verificar quais pontos são atendidos e quais não são atendidos.
- Aprofundar no estudo de mecanismos relacionados ao conceito de Contexto (ver Subseção 3.1.3) com foco na melhoria da interação homem-máquina e da qualidade de respostas geradas pelos sistemas.
- Desenvolver modelos de mapeamento mental direcionados para a melhoria dos mecanismos utilizados em um QAS, por exemplo, permitindo criar modelos lógicos que respondam a perguntas mais complexas.
- Trabalhar as capacidades e oportunidades do pré-processamento de fontes de informação, sendo esta direcionada pela avaliação de como um ser humano avalia e seleciona as informações e como as utiliza para responder a questões.

## REFERÊNCIAS

- BHASKAR, P.; PAL, B. C.; BANDYOPADHYAY, S. **Comparative & Evaluative QA system in tourism domain.** In: Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology (CCSEIT '12). 2012. p. 458-465.
- COMAS, P. R.; TURMO, J.; MÀRQUEZ, L. Sibyl, a factoid question-answering system for spoken documents. **Transactions on Information Systems (TOIS)**, v. 30, n. 3, p. 19:1-19:40, 2012.
- HAKIMOV, S.; TUNC, H.; AKIMALIEV, M.; DOGDU, E. **Semantic question answering system over linked data using relational patterns.** In: Proceedings of the Joint EDBT/ICDT 2013 Workshops (EDBT '13). 2013. p. 83-88.
- KACPRZYK, J.; ZADROZNY, S. **Comprehensiveness and interpretability of linguistic data summaries A natural language focused perspective.** In: 2013 IEEE Symposium on Computational Intelligence for Human-like Intelligence (CIHLI). 2013. p. 33 - 40.
- KALYANPUR, A.; PATWARDHAN, S.; BOGURAEV, B.; CHU-CARROLL, J.; LALLY, A. **Parallel and nested decomposition for factoid questions.** In : Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL '12). 2012. p. 851–860.
- LIU, R.; NYBERG, E. **A phased ranking model for question answering.** In: Proceedings of the 22nd ACM Conference on Information & Knowledge Management (CIKM '13). 2013. p. 79-88.
- WALKE, P. P.; KARALE, S. **Implementation approaches for various categories of question answering system.** In: 2013 IEEE Conference on Information & Communication Technologies (ICT '13). 2013. p. 402 - 407.
- WANG, M. A Survey of Answer Extraction Techniques in Factoid Question Answering. **Association for Computational Linguistics**, v. 1, n. 1, p. 1-14, 2006.

## BIBLIOGRAFIA COMPLEMENTAR

- AGARWAL, A.; RAGHAVAN, H.; SUBBIAN, K.; MELVILLE, P.; LAWRENCE, R. D.; GONDEK, D. C.; FAN, J. **Learning to rank for robust question answering.** In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12). 2012. p. 833-842.
- AHN, C.; LEE, J.; CHOI, B.; PARK, S. **Question answering system with recommendation using fuzzy relational product operator.** In: Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services (iiWAS '10). 2010. p. 855-858.
- BING, S.; LEI, F. **Keyword Extraction Algorithm Based on.** In: 2013 Fifth International Conference on Computational and Information Sciences (ICCIS). 2013. p. 664-665.
- BYKAU, S.; RIZZOLO, F.; VELEGRAKIS, Y. **A query answering system for data with evolution relationships.** In: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD '13). 2013.
- CAO, X.; CONG, G.; CUI, B.; JENSEN, C. S.; YUAN, Q. Approaches to Exploring Category Information for Question Retrieval in Community Question-Answer Archives. **Transactions on Information Systems (TOIS)**, v. 30, n. 2, p. 7:1-7:38, 2012.
- CHAI, J. Y.; ZHANG, C.; BALDWIN, T. **Towards conversational QA automatic identification of problematic situations and user intent.** In: Proceedings of the COLING/ACL on Main Conference Poster Sessions (COLING-ACL '06). 2006. p. 57-64.
- CHUA, T.; HONG, R.; LI, G.; TANG, J. **From text question-answering to multimedia QA on web-scale media resources.** In: Proceedings of the First ACM Workshop on Large-scale Multimedia Retrieval and Mining (LS-MMRM '09). 2009. p. 51-58.

- ELIZABETH, E. R. P.; RAMPRASATH, M.; HARIHARAN, S. **Improving QA processing by semantic reformulation.** In: 2013 International Conference on Computer Communication and Informatics (ICCCI). 2013. p. 1-4.
- FADER, A.; ZETTELEMOYER, L.; ETZIONI, O. **Open question answering over curated and extracted knowledge bases.** In: Proceedings of the 20th ACM International Conference on Knowledge Discovery and Data Mining (KDD '14). 2014. p. 1156-1165.
- FANG, L.; HUANG, M.; ZHU, X. **Question routing in community based QA: incorporating answer quality and answer content.** In: Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics (MDS '12). 2012. p. 1-8.
- FERRUCCI, D.; BROWN, E.; CHU-CARROLL, J.; FAN, J.; GONDEK, D.; KALYANPUR, A. A.; LALLY, A.; MURDOCK, J. W.; NYBERG, E.; PRAGER, J.; SCHLAEFER, N.; WELTY; C. Building Watson An Overview of the DeepQA Project. **AI Magazine**, n. 31(3), p. 59-79, 2010.
- FIGUEROA, A. **Can click patterns across user's query logs predict answers to definition questions?** In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. 2012. p. 99-108.
- GONÇALVES, P. N.; BRANCO, A. **Open-Domain Web-Based List Question Answering with LX-ListQuestion.** In: Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14). 2014. p. 1-6.
- GRAPPY, A.; GRAU, B.; ROSSET, S. **Methods combination and ML-based re-ranking of multiple hypothesis for question-answering systems.** In: Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data (Hybrid2012). 2012. p. 87–96.
- HU, D.; WANG, W.; XIE, N.; CAO, C. **ACQA\_onto An ontology approach for restrain domain question answering system.** In: IET International Conference on Information Science and Control Engineering 2012 (ICISCE 2012). 2012. p. 1-5.

JI, Z.; XU, F.; WANG, B.; HE, B. **Question-answer topic model for question retrieval in community question answering.** In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12). 2012. p. 2471-2474.

KAISSER, M. **Answer sentence retrieval by matching dependency paths acquired from question/answer sentence pairs.** In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL '12). 2012. p. 88–98.

KALYANPUR, A.; PATWARDHAN, S.; BOGURAEV, B.; CHU-CARROLL, J.; LALLY, A. **Fact-Based Question Decomposition for Candidate Answer Re-Ranking.** In: International Conference on Information and Knowledge Management. 2011. p. 2045-2048.

KIM, H.; KIM, K.; LEE, G. G.; SEO, J. **MAYA a fast Question-answering system based on a predictive answer indexer.** In: Proceedings of the workshop on Open-Domain Question Answering (ODQA '01). 2001. p. 1-8.

KIM, S.; KIM, H.-S. **Ontology modeling for provision of semantic based open API information.** In: 15th International Conference on Advanced Communication Technology (ICACT 13). 2013. p. 664-667.

KIRSCHNER, M.; BERNARDI, R. **Towards an empirically motivated typology of follow-up questions: the role of dialogue context.** In: Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL '10). 2010. p. 322-331.

KO, J.; SI, L.; NYBERG, E.; MITAMURA, T. Probabilistic models for answer-ranking in multilingual question-answering. **Transactions on Information Systems (TOIS),** 28, n. 3, 2010. 16:1-16:37.

KOMIYA, K.; ABE, Y.; KOTANI, Y. **Query expansion using mutual information between words in question and its answer.** In: Proceedings of the ITI 2013 35th

- International Conference on Information Technology Interfaces (ITI '13). 2013. p. 289-294.
- LEE, U.; KANG, H.; YI, E.; YI, M.; KANTOLA, J. **Understanding mobile Q&A usage An exploratory study.** In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12). 2012. p. 3215-3224.
- LI, S.; MANANDHAR, S. **Automatic generation of information-seeking questions using concept clusters.** In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers. 2009. p. 93–96.
- LIU, Z.; QIU, X.; CAO, L.; HUANG, X. **Discovering logical knowledge for deep question answering.** In: Proceedings of the 21st ACM Information and Knowledge Management (CIKM '12). 2012. p. 1920-1924.
- LU, Z.; ZHA, H.; YANG, X.; LIN, W.; ZHENG, Z. Sessions, A New Algorithm for Inferring User Search Goals with Feedback. **IEEE Transactions on Knowledge and Data Engineering**, v. 25, n. 3, p. 502-513, 2013.
- MANGU, L.; SOLTAU, H.; KUO, H.-K.; KINGSBURY, B.; SAON, G. **Exploiting diversity for spoken term detection.** In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '13). 2013. p. 8282-8286.
- MANOHAR, C. S. **Natural Language Processing Future.** In: Proceedings of International Conference on Optical Imaging Sensor and Security. 2013. p. 1-3.
- MOLDOVAN, D.; PAŞCA, M.; HARABAGIU, S.; SURDEANU, M. Performance issues and error analysis in an open-domain question answering system. **Transactions on Information Systems (TOIS)**, 21, n. 2, 2013. 133–154.
- NAJMI, E.; HASHMI, K.; KHAZALAH, F.; MALIK, Z. **Intelligent semantic question answering system.** In: 2013 IEEE International Conference on Cybernetics (CYBCONF '13). 2013. p. 255-260.
- OLIVEIRA, A. L. C.; SILVA, E. S.; MACEDO, H. T.; MATOS, L. N. **Brazilian Portuguese speech-driven answering system.** In: Proceedings of the 6th Euro

American Conference on Telematics and Information Systems (EATIS '12). 2012. p. 277-284.

PICCARDI, T.; CONVERTINO, G.; ZANCANARO, M.; WANG, J.; ARCHAMBEAU, C. **Towards crowd-based customer service a mixed-initiative tool for managing Q&A sites.** In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14). 2014. p. 2725-2734.

RAMPRASATH, M.; HARIHARAN, S. **Improving QA performance through semantic reformulation.** In: 2012 Nirma University International Conference on Engineering (NUiCONE). 2012. p. 1-4.

RZESZOTARSKI, J. M.; SPIRO, E. S.; MATIAS, J. N.; MONROY-HERNÁNDEZ, A.; MORRIS, M. R. **Is anyone out there? Unpacking Q&A hashtags on twitter.** In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14). 2014. p. 2755-2758.

SEVERYN, A.; MOSCHITTI, A. **Structural relationships for large-scale learning of answer re-ranking.** In: Proceedings of the 35th international ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR '12). 2012. p. 741-750.

SHARMA, M.; SUPRIYA. **Deep Web Data Extraction Using Query String Formation.** In: 2014 International Conference on Reliability, Optimization and Information Technology (ICROIT 2014). 2014. p. 166-169.

SHEKARPOUR, S.; NGOMO, A.-C. N.; AUER, S. **Question answering on interlinked data.** In: Proceedings of the 22nd international conference on World Wide Web (WWW '13). 2013. p. 1145-1155.

SHTOK, A.; DROR, G.; MAAREK, Y.; SZPEKTOR, I. **Learning from the past answering new questions with past answers.** In: Proceedings of the 21st international conference on World Wide Web (WWW '12). 2012. p. 759-768.

- SIDDIQI, R. **A Database Design for Complex Linguistic Data in Collaborative Web.** In: 2013 5th International Conference on Information & Communication Technologies (ICICT '13). 2012. p. 1 - 7.
- SMUCKER, M. D.; ALLAN, J.; DACHEV, B. **Human question answering performance using an interactive document retrieval system.** In: Proceedings of the 4th Information Interaction in Context Symposium (IIIX '12). 2012. p. 35-44.
- STERBINI, A.; TEMPERINI, M. **Analysis of open answers via mediated peer-assessment.** In: 2013 17th International Conference System Theory, Control and Computing (ICSTCC '13). 2013. p. 663-668.
- STERBINI, A.; TEMPERINI, M. **OpenAnswer, a framework to support teacher's management of open answers through peer assessment.** In: 2013 IEEE Frontiers in Education Conference. 2013. p. 164-170.
- SUN, M.; CHAI, J. Y. **Towards intelligent QA interfaces Discourse processing for context questions.** In: Proceedings of the 11th international conference on Intelligent User Interfaces (IUI '06). 2006. p. 163-170.
- TUTOS, A.; MOLLA, D. **A study on the use of search engines for answering clinical questions.** In: Proceedings of the Fourth Australasian Workshop on Health Informatics and Knowledge Management. 2010. p. 61-68.
- VIGO, M.; JAY, C.; STEVENS, R. **Design insights for the next wave ontology authoring tools.** In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14). 2014. p. 1555-1558.
- WONG, W.; CAVEDON, L.; THANGARAJAH, J.; PADGHAM, L. **Mixed-initiative conversational system using question-answer pairs mined from the web.** In: Proceedings of the 21st ACM international conference on Information and Knowledge Management (CIKM '12). 2012. p. 2707-2709.
- WU, Y.; KASHIOKA, H. **An Unsupervised Model of Exploiting the Web to Answer Definitional Questions.** In: International Joint Conferences on Web Intelligence and Intelligent Agent Technologies. 2009. p. 28 - 33.

XIAOMING, L.; JIAN, Z. **Semantic Unit Based Question Representation**. In: 2012 International Conference on Industrial Control and Electronics Engineering (ICICEE '12). 2012. p. 1595-1598.

XUE, X.; JEON, J.; CROFT, W. B. **Retrieval models for question and answer archives**. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR '08). 2008. p. 475-482.

YAHYA, M.; BERBERICH, K.; ELBASSUONI, S.; RAMANATH, M.; TRESP, V.; WEIKUM, G. **Natural language questions for the web of data**. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL '12). 2012. p. 379–390.

YIN, B.; YUE, X.; CAI, D.; ZHANG, G. **Analysis and Evaluation of Terminology Translation Consistency in Scientific and Technical Literature**. In: 2013 International Conference on Asian Language Processing (IALP '13). 2013. p. 83-86.

ZHOU, G.; CAI, L.; ZHAO, J.; LIU, K. **Phrase-based translation model for question retrieval in community question answer archives**. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT '11). 2011. p. 653–662.

ZHOU, G.; LIU, K.; ZHAO, J. **Joint relevance and answer quality learning for question routing in community QA**. In: Proceedings of the 21st ACM international conference on Information and Knowledge Management (CIKM '12). 2012. p. 1492-1496.

ZHOU, Z.; SELLAMI, M.; GAALOUL, W.; BARHAMGI, M.; DEFUDE, B. Data Providing Services Clustering and Management for Facilitating Service Discovery and Replacement. **IEEE Transactions on Automation Science and Engineering**, v. 10, n. 4, p. 1131-1146, 2013.

## APÊNDICE A – Detalhamento das Implementações

Este apêndice inclui detalhes de processamentos por QAS analisado no Capítulo 2.

### IMPLEMENTAÇÃO 1

A interpretação da questão envolve a identificação de 44 tipos de categoria (ex. local, data, nome próprio) e 110 subcategorias (ex. aeroporto, montanha, pessoa). Os pronomes interrogativos (ex. o que, quanto) são analisados para identificar o tipo genérico de resposta a ser dada. Um analisador semântico baseado em papéis, como palavra-chave, também auxilia na função de interpretação. Uma chave de três níveis é formada, sendo sua estrutura: “tipo genérico: categoria: subcategoria”.

As informações obtidas fornecem dados para compor *queries* que são utilizadas por agentes de busca com o objeto de coletar informações relevantes que possam conter a resposta desejada. A busca integra múltiplos agentes, construídos por terceiros, como Indri<sup>5</sup> e Bing Search API. A busca por informações ocorre tanto localmente como em fontes baseadas na WEB.

Os trechos coletados pelos agentes de busca são analisados para identificação de entidades. As entidades são comparadas com as informações obtidas durante a etapa de interpretação da pergunta com o objetivo de filtrar os trechos que apresentem maior potencial para gerar a resposta desejada. Oito extratores de entidades são integrados dentro do modelo, sendo estes uma união de extratores desenvolvidos por terceiros com adaptações desenvolvidas pelos autores.

Todas as informações geradas são analisadas dentro de uma visão sistêmica, sendo utilizado para isto um gráfico sistêmico de objetos. O gráfico sistêmico de objetos é integrado a toda cadeia de processamento e armazena características relevantes sobre entradas, modelos, saídas, nós de processamento e dependências, além de

---

<sup>5</sup> <http://www.lemurproject.org/indri/>

valores utilizados no modelo de ranqueamento. Este gráfico tem como objetivo melhorar o processo de afunilamento (seleção e ranqueamento) e permite que o sistema aprenda como maximizar a precisão dos resultados calibrando o peso das características avaliadas para os objetos.

O ranqueamento é parte fundamental do processo, sendo que a resposta posicionada como primeira é a dada como resposta final. Ele mescla duas técnicas: Cascata e Votação de Relevância. O modelo Cascata é aplicado para extrair os atributos gerados em cada ponto do gráfico de objetos (entrada, dependência e saída) combinando-os linearmente para gerar uma pontuação normalizada. O modelo de Votação de Relevância leva em consideração que uma resposta deve ser analisada comparando sua relevância frente à relevância de respostas vizinhas e utiliza em conjunto uma função que trabalha a similaridade dos resultados.

## **IMPLEMENTAÇÃO 2**

Durante a interpretação da pergunta o sistema analisa uma série de componentes que permitem mapear o que o usuário deseja obter como resposta. Os componentes são: Contexto, Expressão Comparativa ou Aferitiva, Entidade, Modo de Comparaçāo e Restrição. O Contexto representa as demandas e necessidades do usuário. A Expressão Comparativa ou Aferitiva (*Comparative or Evaluative Expression*, CEE) identifica os termos de comparação ou aferição das Entidades. A Entidade representa os objetos a serem comparados. O Modo de Comparaçāo representa os tipos possíveis de comparação. A Restrição representa critérios que guiam a comparação identificando restrições relacionadas às preferências do usuário.

O processo de identificação dos componentes envolve o que se chama de Atributos Comparativos e Aferitivos (*Comparative and Evaluative Features*, CEF). Os autores identificam 17 tipos, como Local Para, Local De, Local Indispensável, Tempo de Ida, Tempo Limite, Detalhe do Viajante, Detalhe do Grupo, Orçamento, Adjetivo Modificador etc. Com os CEF em mãos, é identificado o Contexto da pergunta. O Contexto apresenta sete classes: Itinerário, Acomodação, Transporte, Passeios, Tempo e Custo. Baseados no CEF e no Contexto, são extraídas as informações de Entidades e Restrições seguindo regras pré-definidas na base de conhecimento

específico do domínio. A base de conhecimento específico do domínio é construída manualmente visando permitir que o sistema entenda como um ser humano interpretaria as informações apresentadas. Ela apresenta os seguintes propósitos: 1 – Extração de atributos relacionados ao Contexto, Expressões Comparativas e Aferitiva, Entidade e Restrições, 2 – Predição de atributos não especificados, 3 – Decomposição da entidade em propriedades que caracterizem a entidade e 4 – Atribuição de importância aos atributos.

Por fim, são obtidas as Expressões Comparativas ou Aferitivas e o Modo de Comparaçāo. As Expressões Comparativas ou Aferitivas são extraídas analisando os adjetivos e advérbios. O Modo de Comparaçāo é classificado em três tipos: Comparação Genérica, Comparação Comparativa e Comparação Superlativa. A Comparação Genérica ocorre, normalmente, quando uma questão é respondida com sim/não, como no exemplo: São Paulo é uma boa cidade para ser visitada? A Comparação Comparativa é identificada quando na pergunta são relacionadas duas ou mais entidades para serem comparadas, como no exemplo: Qual cidade é melhor para se visitar, Rio de Janeiro ou São Paulo? A Comparação Superlativa é identificada quando na pergunta se delimita entidades relacionadas a um termo específico, como no exemplo: Qual cidade do Brasil é a melhor para ser visitada?

Interpretada a pergunta, é iniciada a busca por dados que possam conter a resposta desejada. O ponto turístico, que pode ser específico (ex. Monte Everest) ou genérico (ex. Montanha), é fundamental para direcionar a busca, sendo utilizado como guia para ler as informações existentes na tabela Documentos Indexados. Esta é uma das três tabelas locais existentes utilizadas durante a busca por informações, sendo estas: 1) Localização Hierarquizada, contendo dados de cidade, estado, país, latitude, longitude, população, língua, tipo de turismo etc., 2) Documentos Indexados, contendo dados de endereço de documentos, indexação direta e indexação indireta e 3) Detalhes de Aeroportos e Ferrovias, contendo dados de cidade, aeroporto, código IANA, estação de trem, código de estação de trem. WikiTravel, Lonely Planet, Kayak e Google Maps são as fontes que proveem informações para o sistema. Estas fontes têm a característica de serem HTML

estáticas ou construídas sobre o conceito de *live feed*. Utilizam HTML tags especiais, agrupamento de informações ou RSS feed que auxiliam na busca.

Coletados os trechos, estes são tratados com o objetivo de serem identificados objetos de entidade e informações associadas a estes objetos. Com estes em mãos, são mapeados atributos seguindo regras pré-estabelecidas na base de conhecimento específico do domínio. Os atributos são valorados, também com base em regras existentes na base de conhecimento específico do domínio. As pontuações dadas aos atributos de cada objeto de entidade dos trechos coletados são somadas de forma ponderada. O sistema compara as pontuações obtidas e consideram os cinco objetos de entidades de maior pontuação como respostas.

### IMPLEMENTAÇÃO 3

Durante a interpretação da pergunta, o sistema trabalha para reconhecer e tipificar os múltiplos fatos da questão. Para isto, é utilizado o componente *Decomposition Recognizer*, que utiliza regras e padrões léxico-sintáticos para quebrar a pergunta em fatos independentes, transformando estes fatos em subquestões. Há três padrões genéricos de quebra utilizados: Sub-Árvores Independentes, Unidades Combináveis e Segmento com Qualificadores. No padrão Sub-Árvores Independentes a derivação da questão em subquestões ocorre de forma direta. O padrão Unidades Combináveis é utilizado para questões das quais a derivação das subquestões depende da combinação de elementos existentes dentro da pergunta. O padrão Segmentos com Qualificadores é utilizado para questões das quais a derivação deve considerar um modificador que qualifique o foco da pergunta, como “o primeiro”. Seguem exemplos de perguntas que se encaixam em cada um dos padrões:

- Sub-Árvores Independentes: Quem é o físico biografado em American Prometheus e que morreu em 1967?
- Unidades Combináveis: Quem é o homem do qual a filha, para honrar seu trabalho, adotou o nome de Maria Celeste quando se tornou freira em 1616?

- Segmento com Qualificadores: Ganhando em 1965 e 1966, quem foi o primeiro homem a obter duas vezes consecutivas o título do torneio de golfe Master?

As subquestões geradas recebem tratamentos heurísticos para eliminar as que não se aplicam ao caso analisado. Descartam-se, por exemplo, subquestões que não contenham ao menos uma entidade, data, período ou citação. Também se descartam subquestões que acabam sendo quase que uma reprodução da questão. Como a quebra em subquestões pode deixar algumas informações importantes de fora, estas subquestões são tratadas por meio do componente *Question Rewriters*, garantido que informações chave sejam mantidas juntas às subquestões. O tipo de dependência entre as subquestões também é definido, existindo dois tipos: Paralelo ou Aninhado. A dependência Paralela define que as subquestões são tratadas de forma independente. A dependência Aninhada define que as subquestões são processadas conforme ordem lógica determinada por qual subquestão “contém” a outra.

A busca por trechos é realizada pelo QAS Watson, da IBM. O sistema processa individualmente cada uma das subquestões e gera uma lista de trechos ranqueada conforme índices de confiança obtidos. Porém, há diferenças no processo de busca de informação entre as subquestões de dependência Paralela e Aninhada. No caso da dependência Paralela, as diferentes subquestões são processadas de forma paralela. No caso da dependência Aninhada, é utilizado o conceito *feedback loop*. Como ocorre na dependência Paralela, cada subquestão é tratada individualmente durante a coleta de informações, porém isto ocorre de forma sequencial, não paralela. Primeiro são coletados trechos para a subquestão “contida” e estes ranqueados. A lista ranqueada é processada pelo componente *Question Rewriters* para modificar a subquestão “contém”, compondo nova subquestão que é utilizada para uma nova coleta de informações. Por fim, tanto para o processamento de subquestões paralelas quanto aninhadas, entra no jogo o componente *Answer Synthesis and Re-ranking*, que realiza a recombinação dos trechos coletados, os sintetizando e re-rankeando.

É importante ressaltar que, além de processar as subquestões, o QAS Watson também processa a pergunta sem decompô-la. Este fato é destacado aqui porque, para a definição da resposta final, as informações obtidas pelo procedimento de Kalyanpur, Patwardhan, Boguraev, Chu-Carroll e Lally (2012) são cruzadas com as informações obtidas via processo original de processamento da pergunta do QAS Watson.

O componente *Answer Synthesis and Re-ranking* trabalha de forma diferente as listas geradas pela coleta dos trechos quando consideradas as dependências Paralela e Aninhada. No caso da dependência Paralela, as listas geradas pelas subquestões são unificadas, sendo a nova lista cruzada com a lista gerada pelo tratamento da pergunta não decomposta. Este cruzamento serve para rever os índices de confiança conforme peso de atributos heurísticos, sendo a resposta melhor ranqueada selecionada como resposta final. No caso da dependência Aninhada, a resposta é selecionada pelo cruzamento da lista obtida por meio do *feedback loop* com a lista gerada pelo tratamento da pergunta não decomposta. A resposta final é selecionada através da comparação entre os índices de confiança obtidos no tratamento das subquestões versus os índices de confiança obtidos no tratamento da pergunta não decomposta, sendo escolhida a que apresentar o maior índice de confiança.

#### IMPLEMENTAÇÃO 4

Interpreta-se a pergunta com o objetivo de traduzi-la para um padrão RDF de três elementos alinhado à base ontológica da DBpedia. Para isto, primeiro a pergunta é decomposta por meio de ferramentas de análise de linguagem natural contidas na Stanford CoreNLP (SCNLP)<sup>6</sup>, chegando-se a uma árvore de dependência que inclui *Part-Of-Speech Tags* (POST) como sujeito, objeto, tipos de objeto, modificadores parciais, verbos de ligação, entre outros. Depois a árvore é analisada

---

<sup>6</sup> <http://nlp.stanford.edu/software/corenlp.shtml>

recursivamente por uma ferramenta de extração de elementos com o objetivo de compor conjuntos de três elementos e determinar a relação entre estes. Cada conjunto de elementos é dividido em Sujeito, Predicado e Objeto e seu preenchimento, realizado pela ferramenta de extração de elementos, é convertido para os padrões existentes na base ontológica da DBpedia. Padrão, frequência e similaridade são técnicas utilizadas no alinhamento com a base ontológica da DBpedia.

Durante a interpretação da pergunta, também é definido o tipo de resposta esperada, utilizando a seguinte relação entre pronome interrogativo e tipo de resposta: Quem = Pessoa, Organização, Empresa; Onde = Local; Quando = Data; Quanto = Número. No caso de questões que utilizam Qual, não é necessário haver a checagem do tipo de resposta devido à importância que o substantivo assume neste tipo de pergunta.

Com base nas informações obtidas, são estruturadas SPARQL *queries* para coletar informações na DBpedia. Este processo leva em consideração todas as possíveis combinações entre os elementos encontrados, tomando como base a lista RDF já alinhada à ontologia da DBpedia. A cada conjunto de três elementos trabalhado pelas *queries* é atrelado uma pontuação que toma como base a frequência verificada pelos predicados.

O sistema checa o alinhamento entre as informações coletadas e o tipo de resposta esperada. Os candidatos que estão alinhados são ranqueados conforme a pontuação de frequência do predicado. O candidato que aparece na primeira posição do ranqueamento é considerado como sendo a resposta final da pergunta.

## **IMPLEMENTAÇÃO 5**

A interpretação da pergunta, que deve ser relacionada a um fato, identifica o tipo de resposta esperada. São 53 tipos, sendo que a definição de qual utilizar ocorre por meio de um modelo de classificação *perceptron* multiclasse e da atribuição de uma série de características léxicas, sintáticas e semânticas. Definidos os tipos de resposta, estas são reclassificadas entre sete Entidades reconhecíveis pelo

componente *Named Entity Recognition and Classification* (NERC): Pessoa, Organização, Local, Medida, Número, Data e Hora. Também é identificado um conjunto de palavras-chave que será utilizado nas *queries* de coleta de informação. Atrelado às palavras-chave, é designados um relevância heurística, sendo a pontuação fixa (ex. Adjetivos = 5, Entidades = 8).

A busca por informações trabalha com dois modelos de coleta, ambos utilizando o conjunto de palavras-chave identificado anteriormente. O primeiro é um modelo baseado em texto e o segundo adaptado para transcrições realizadas por meio de mecanismos automáticos (*Automatic Speech Recognition*, ASR). O modelo baseado em texto processa o conjunto de palavras-chave identificadas no módulo anterior estruturando *queries* dinâmicas (*Dynamic Query Relaxation Procedure*) que trabalham de forma a ajustar iterativamente o número de palavras-chave e a proximidade com que estas devem ser encontradas dentro da fonte consultada. A *query* é ajustada até a qualidade da coleta se tornar satisfatória. A ferramenta de extração de informação utilizada é a Lucene IR (Apache). O modelo de coleta de informação adaptado para transcrições utiliza o mecanismo de extração PHAST (*PHonetic Alignment Search Tool*), sendo este um mecanismo SDR (*Spoken Document Retrieval*). Este modelo utiliza similaridade fonética para encontrar as palavras-chave, melhorando a interpretação dos trechos de documentos obtidos com a utilização de mecanismos ASR. Os trechos coletados são pontuados conforme frequência dos termos encontrados.

Os trechos coletados dentro dos dois modelos de coleta de informação são analisados com o objetivo de serem identificadas Entidades. Este procedimento passa pelo NERC e é separado em duas fases. A primeira fase é a *Entity Recognition Model*, que trabalha com a técnica BIO *tagging scheme*. Cada palavra do trecho é classificada com B, I ou O. B (*Beginning*) ocorre quando é início de uma nova Entidade. I (*Inside*) ocorre quando outra palavra está dentro da Entidade. O (*Outside*) ocorre quando uma palavra não está atrelada a uma Entidade. A segunda fase chama-se *Entity Classification Model*, que trabalha as tags classificando-as dentro dos padrões pré-definidos de Entidades (Pessoa, Organização, Local, Medida, Número, Data e Hora).

As Entidades encontradas nos trechos coletados são cruzadas com as identificadas durante a interpretação da pergunta. Os trechos que potencialmente contenham a resposta são ranqueados por meio de uma composição de pontuações. Os dois principais modelos de pontuação apresentam bases heurísticas, um medindo a distância e densidade das palavras-chave e outro tratando a relação de similaridade sintática entre trechos e pergunta. A resposta que aparece na primeira posição do ranqueamento é considerada a final.

## APÊNDICE B – Reuso e Métricas

Há uma série de modelos e fontes de dados desenvolvidos por terceiros que podem ser utilizados na construção de um QAS. As implementações expostas no Capítulo 2 utilizaram alguns deles. Os autores também empregaram algumas métricas para avaliar as suas propostas. Neste apêndice são mapeadas estas informações em tabelas, conforme explicado abaixo:

1. *Modelos*: mapeia os modelos desenvolvidos por terceiros que foram integrados na proposta de implementação desenvolvida por cada autor.
2. *Fontes de Dados*: mapeia as fontes de dados utilizadas tanto para a coleta de informação com intenção de gerar a resposta quanto para testar e treinar os sistemas.
3. *Métricas de Desempenho*: descreve as métricas utilizadas pelos autores para avaliar o desempenho de suas ferramentas.

Os termos estão escritos conforme se encontram nos artigos analisados para não criar problemas de interpretação, especialmente por muitos serem específicos e não terem tradução consolidada.

**Tabela A. Modelos**

Artigo	Modelos
<b>A Phased Ranking Model For Question Answering</b>	<ul style="list-style-type: none"> <li>→ OpenEphyra Framework and NER For Open Domain Question Answering, <i>Ephyra</i></li> <li>→ OpenNLP NER, <i>Apache OpenNLP</i></li> <li>→ Stanford NER, <i>Stanford</i></li> <li>→ Indri Search Engine, <i>Lemur Project</i></li> <li>→ Bing Search API, <i>Bing</i></li> <li>→ Wikipedia Search API, <i>Wikipedia</i></li> <li>→ Yahoo! Answers API, <i>Yahoo</i></li> <li>→ Twitter Search API, <i>Twitter</i></li> <li>→ Queries Methods, <i>D. Metzler and W. B. Croft. A markov random</i></li> </ul>

	<i>field model for term dependencies. SIGIR '05</i>
<b>Comparative &amp; Evaluative QA System In Tourism Domain</b>	Artigo não apresenta informações relacionadas a este ponto.
<b>Parallel And Nested Decomposition For Factoid Questions</b>	<p>→ Watson DeepQA Project, <i>D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J.W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty. 2010. Building Watson: An Overview of the DeepQA Project. AI Magazine</i>, 31(3):59–79, Fall.</p> <p>→ Weka's Logistic Regression Algorithm, <i>Witten and E. Frank. 2000. Data Mining – Practical Machine Learning Tools and Techniques with Java Implementations. Morgan-Kaufmann, San Francisco, CA.</i></p>
<b>Semantic Question Answering System Over Linked Data Using Relational Patterns</b>	<p>→ Stanford CoreNLP Library (SCNLP), <i>Stanford</i></p> <p>→ WordNet::Similarity, <i>Pedersen, T., Patwardhan, S., &amp; Michelizzi, J. 2004. WordNet:: Similarity: measuring the relatedness of concepts. In Demonstration Papers at HLT-NAACL 2004 (pp. 38-41). Association for Computational Linguistics.</i></p> <p>→ WordNet Searching API (JAWS), <i>Bobby B. Lyle School of Engineering at Southern Methodist University</i></p> <p>→ PATTY, <i>Nakashole, N., Weikum, G., &amp; Suchanek, F. 2012. PATTY: A Taxonomy of Relational Patterns with Semantic Types. Empirical Methods in Natural Language Processing and Natural Language Learning (EMNLP 2012)</i></p>
<b>Sibyl, A Factoid Question-Answering System For Spoken Documents</b>	<p>→ Multiclass Perceptron Classifier, <i>LI, X. AND ROTH, D. 2005. Learning question classifiers: The role of semantic information. J. Natural Lang. Eng.</i></p> <p>→ TnT tagger, <i>BRANTS, T. 2000. TnT: A statistical part-of-speech tagger. In Proceedings of the 6th Conference on Applied Natural Language Processing. ACL.</i></p> <p>→ Carnegie Mellon Pronouncing Dictionary, <i>Carnegie Mellon University</i></p> <p>→ SoundChange Package, <i>Perl's Library</i></p> <p>→ RelaxCor, <i>SAPENA, E., PADRO, L., AND TURMO, J. 2010. A global relaxation labeling approach to coreference resolution. In</i></p>

	<i>Proceedings of the COLING Conference.</i>
--	--

**Tabela B. Fontes de Dados**

Artigo	Fontes Utilizadas para Coleta de Dados, Treinamento ou Testes
<b>A Phased Ranking Model For Question Answering</b>	<ul style="list-style-type: none"> <li>→ AQUAINT Downloaded Corpus, <i>Linguistic Data Consortium (LDC)</i></li> <li>→ Wikipedia Downloaded Corpus, <i>Wikipedia</i></li> <li>→ TREC QA Datasets, <i>Text Retrieval Conference</i></li> </ul>
<b>Comparative &amp; Evaluative QA System In Tourism Domain</b>	<ul style="list-style-type: none"> <li>→ Wikitravel WEB Page, <i>Wikitravel</i></li> <li>→ LonelyPlanet WEB Page, <i>Lonely Planet</i></li> <li>→ Kayak WEB Page, <i>Kayak</i></li> <li>→ Google Map, <i>Google</i></li> <li>→ Ask Markopolio Section, <i>OutlookTraverller Magazine</i></li> </ul>
<b>Parallel And Nested Decomposition For Factoid Questions</b>	<ul style="list-style-type: none"> <li>→ Jeopardy! Quiz, <i>j-archive.com</i></li> </ul>
<b>Semantic Question Answering System Over Linked Data Using Relational Patterns</b>	<ul style="list-style-type: none"> <li>→ Dbpedia, <i>Dbpedia</i></li> <li>→ QALD-2 (Question Answering over Linked Data) Workshop, <i>ESWC 2012 Extended Semantic Web Conference</i></li> </ul>
<b>Sibyl, A Factoid Question-Answering System For Spoken Documents</b>	<ul style="list-style-type: none"> <li>→ European Parliament Plenary Sessions (EPPS), <i>TC-STAR project financed by European Commission</i></li> <li>→ CoNLL Shared Task Collection, <i>Special Interest Group on Natural Language Learning of the Association for Computational Linguistics</i></li> </ul>

**Tabela C. Métricas**

Artigo	Métricas de Desempenho
<b>A Phased Ranking Model For Question Answering</b>	<ul style="list-style-type: none"> <li>→ TOP@1: quantidade de respostas consideradas como corretas e classificadas em primeiro lugar no ranqueamento dividido pela quantidade total de perguntas realizadas</li> </ul>

	<p>→ MRR@5: mediana das médias calculada sobre ranqueamentos considerando as 5 primeiras respostas da lista</p> <p>→ Recall@10: pontua como 1 caso a resposta correta esteja entre as 10 primeiras no ranqueamento e 0 caso não seja encontrada</p>
<b>Comparative &amp; Evaluative QA System In Tourism Domain</b>	<p>→ Precision: são dadas notas de 0-5 por pessoas e estas são utilizadas conforme equação abaixo</p> $\text{Precision} = \frac{\sum_{i=1}^{\text{no. of questions}} \left( \left( \sum_{j=1}^{10} S_{ij} \times W_j \right) \div \left( 10 \times \sum_{j=1}^{10} W_j \right) \right)}{5 \times \text{no. of Input}} \quad (1)$ <p>where, <math>S_{ij}</math> = score of <math>j^{\text{th}}</math> answer of <math>i^{\text{th}}</math> question and <math>W_j</math> = weight for <math>j^{\text{th}}</math> answer.</p> <p>→ Recall: são dadas notas de 0-5 por pessoas e estas são utilizadas conforme equação abaixo</p> $\text{Recall} = \frac{\sum_{i=1}^{\text{no. of questions}} r_i}{5 \times \text{no. of Input}} \quad (2)$ <p>where, <math>r_i</math> = score given for <math>i^{\text{th}}</math> question.</p>
<b>Parallel And Nested Decomposition For Factoid Questions</b>	<p>→ Decomposition Accuracy: artigo não detalha as regras de cálculo, mas separa este indicador em End-to-End Accuracy e Decomposable Q Accuracy.</p>
<b>Semantic Question Answering System Over Linked Data Using Relational Patterns</b>	<p>→ Precision: calcula o percentual de respostas consideradas corretas em relação ao total de respostas geradas por perguntas corretamente processadas pelo sistema.</p> <p>→ Recall: calcula o percentual de perguntas corretamente processadas pelo sistema em relação ao total de perguntas realizadas</p>
<b>Sibyl, A Factoid Question-Answering System For Spoken Documents</b>	<p>→ Mean reciprocal rank (MRR): calcula a média de pontuação, sendo que esta pontuação representa a posição em que se encontra a resposta correta dentro de uma lista de potenciais respostas.</p> <p>→ Accuracy: Fração de respostas corretas encontradas na primeira posição considerando listas de 5 posições.</p>

**APÊNDICE C – Alternativas Validadoras por Implementação**

Sistema	Resposta: Item 1	Resposta: Item 2	Resposta: Item 3	Resposta: Item 4	Resposta: Item 5
IMPLEMENTAÇÃO 1	- Foto	- Não é necessário	- Não é necessário	- Não é necessário	- Sim, obrigatoriamente - Não é necessário
IMPLEMENTAÇÃO 2	- Comparação - Aferição	- Sim, obrigatoriamente - Não é necessário	- Não é necessário	- Não é necessário	- Não é necessário
IMPLEMENTAÇÃO 3	- Foto - Foto Compasto	- Não é necessário	- Não é necessário	- Não é necessário	- Sim, obrigatoriamente - Não é necessário
IMPLEMENTAÇÃO 4	- Foto	- Não é necessário	- Não é necessário	- Não é necessário	- Não é necessário
IMPLEMENTAÇÃO 5	- Foto	- Não é necessário	- Não é necessário	- Sim, obrigatoriamente - Não é necessário	- Sim, obrigatoriamente - Não é necessário